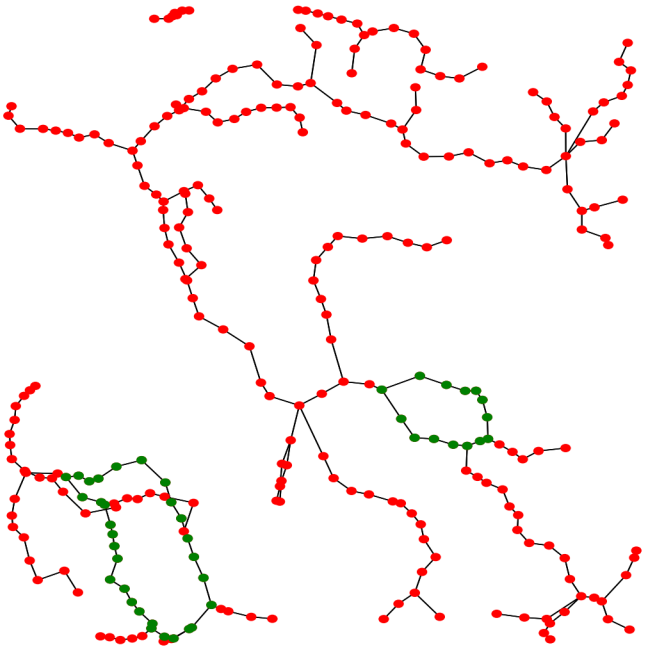
UNIVERSITY OF COPENHAGEN
FACULTY OF SCIENCE



PhD Thesis

Kasra Arnavaz

Segmentation of the Tubular Network in the Pancreas

Department of Computer Science

Advisors: Aasa Feragen, Pia Nyeng, and Oswin Krause

Handed in: October 14, 2021

University of Copenhagen
Faculty of Science
Department of Computer Science

Segmentation of the Tubular Network in the Pancreas

Kasra Arnavaz

Supervisors: Aasa Feragen
Pia Nyeng
Oswin Krause

This thesis has been submitted to the PhD School of The Faculty of Science,
University of Copenhagen, on October 14, 2021

Abstract

Stem cell therapy presents new and exciting opportunities for health care. Its success relies on having an in-depth understanding of the mechanisms which promote cells to take on specialised tasks. With this knowledge, stem cells can be engineered into desired target cells to replace damaged cells. One crucial cell type is the β-cell in the pancreas which produces insulin. Knowing which cues direct the pancreatic cells to turn into β-cells can help patients with type 1 diabetes.

Another function of the pancreas is the production of digestive enzymes which are transported through its tubular network. During embryonic stage, these ductal tubes remodel from a web-like structure to one resembling a tree. It has been hypothesized that this structural change might be a cue which guides pancreatic cells to become β-cells.

This thesis provides the tools to investigate this hypothesis through computational analysis. It is thus required to have segmentations of the tubular structure and extract their topology from the live imaging 3D confocal microscopy. The low signal-to-noise ratio in these images makes attaining large amounts of annotations difficult and costly. On the other hand, deep learning models thrive when loads of labeled data are available.

Active learning is the setting where only the most informative samples are annotated by an expert and added to the training set. However, current sampling strategies are not warranted to perform better than random selection of samples in general. This hinders the utility of active learning for practical applications. We adopt a Bayesian framework and propose selecting samples that contain maximum mutual information between model parameters and labels. We show that for logistic regression model, this criterion is beneficial if the data is roughly linearly separable. For our application, we need more complex models which often render entropy computation intractable.

For a reliable test of the biological hypothesis, we need an accurate measure of segmentation uncertainty. We study established probabilistic segmentation models to see if their estimated uncertainties capture segmentation error. We also investigate the utility of these uncertainties for active learning.

Furthermore, segmentation models are trained for pixel-wise performance, while our application puts an emphasis on topological accuracy. In particular, tubular loops are of major significance. To that end, we derive a topological score function, which decomposes into loop and component subscores, measuring the topological similarity between a target graph and a predicted graph. This score function can be used to encourage topological consistency through model selection. Lastly, loops are tracked over time to not only study their morphological changes but also reduce the number of false positive loops.

Resumé

Stamcelleterapi giver nye og spændende muligheder for sundhedssystemet. Stamcelleterapiens succes afhænger af at have en indgående forståelse af de mekanismer, der fremmer celler til at påtage sig specialiserede opgaver. Med denne viden kan stamceller manipuleres til ønskede celler som erstatning for beskadigede celler. En vigtig celletype er β-cellen i bugspytkirtlen, der producerer insulin. Viden om hvilke tegn der transformerer bugspytkirtelcellerne om til β-celler, kan hjælpe patienter med type 1-diabetes.

En anden funktion af bugspytkirtlen er produktionen af fordøjelsesenzymer, der transporteres gennem dets rørformede netværk. Under embryonalt stadium ombygger disse kanalrør fra en banelignende struktur til en, der ligner et træ. Det er blevet postuleret, at denne strukturelle ændring kan være et tegn, der guider bugspytkirtelceller til at blive β-celler.

Denne afhandling giver værktøjer til at undersøge denne hypotese gennem beregningsanalyse. Det er således påkrævet at have segmenteringer af den rørformede struktur og udtrække deres topologi fra realtids 3D -konfokalmikroskopi billeder. På grund af det lave signal-til-støj-forhold i disse billeder er det svært og dyrt at opnå store mængder annotationer. På den anden side trives deep learning -modeller, når der er masser af annoterede data tilgængelige.

Aktiv læring er hvor kun de mest informative prøver annoteres af en ekspert og tilføjes til træningssættet. Nuværende prøveudtagningsstrategier er imidlertid ikke sikret at performe bedre end tilfældigt udvalg af prøver generelt. Dette forhindrer brugen af aktiv læring til praktiske anvendelser. Vi adopterer en Bayesiansk ramme og foreslår at vælge prøver, der indeholder maksimal gensidig information mellem modelparametre og etiketter. Vi viser, at dette kriterium er fordelagtigt for den logistiske regressionsmodel, hvis dataene er nogenlunde lineært adskillelige. Til vores applikation har vi brug for mere komplekse modeller, som ofte gør entropi -beregning uhåndterlig.

For en pålidelig test af den biologiske hypotese har vi brug for et nøjagtigt mål for segmenteringsusikkerhed. Vi studerer etablerede probabilistiske segmenteringsmodeller for at se, om deres estimerede usikkerheder fanger segmenteringsfejl. Vi undersøger også nytten af disse usikkerheder for aktiv læring.

Endvidere er segmenteringsmodeller uddannet til pixelvis ydelse, mens vores applikation lægger vægt på topologisk nøjagtighed. Især er rørformede ringe af stor betydning. Til dette formål udleder vi en topologisk score -funktion, som nedbrydes til ring- og komponent delscore og måler den topologiske lighed mellem en ønsket graf og en predikteret graf. Denne score -funktion kan bruges til at tilskynde topologisk konsistens gennem modelvalg. Endelig spores loops over tid for ikke kun at studere deres morfologiske ændringer, men også reducere antallet af falsk-positive ringe.

Acknowledgements

Looking back at the past three years, I cannot but feel a deep contentment. It is no exaggeration to say that this PhD has been a turning point in my life. I have grown beyond my expectations, both professionally and personally.

I would like to begin by thanking my main supervisor, Aasa. Her keen insights and unreserved support are what made this project possible. She has shown me the way when I was lost and pointed out the positives when I couldn't find them. Her meticulous attention has often opened my eyes to view problems from new perspectives. I am deeply grateful for all her efforts to make feel at home in Denmark. I would like to thank Pia for giving me access to the data she has painstakingly collected. Her thoughtful comments and guidance have given this thesis a sense of direction, and her patience in teaching me biology has turned the topic into one of my favorites. I would like to thank Oswin who has always been there for me. Be it a coding problem or a theoretical discussion, his help has been indispensable to me. I also like to pay tribute to the memory of David MacKay—a man I never met, yet feel so close to. He has been a true inspiration to me and has shaped my academic knowledge beyond measure. Last but not least, none of this would have been possible without the financial support of Novo Nordisk Foundation.

I have had the privilege to be a part of DanStem during my PhD. I would like to thank everyone at Semb Lab who have not only made me feel welcomed but also have shown great interest in my work. In particular, I am indebted to Jelena for her hard work in providing me with thorough annotations. I also thank Silja for sharing her work and her constructive feedback.

Lastly, I would like to thank my family in Iran, who I miss dearly; my parents whose love is what keeps me going, and my sister who has cheered me during hard times.

Contents

Abstract **i**

Resumé **iii**

Acknowledgements **v**

1 Biological Motivation **1**
1.1 Developmental Biology 1
1.2 Pancreas Development 3

2 Problem Statement **7**

3 Machine Learning Background **11**
3.1 Bayesian Inference 11
3.1.1 Model Fitting 12
3.1.2 Model Comparison 14
3.2 Information Theory 16
3.3 Active Learning 22

4 Summary and Future Work **25**

Bibliography **27**

List of Publications **31**

A Bayesian Active Learning for Maximal Information Gain on Model Parameters **33**

B Is segmentation uncertainty useful? **51**

C U-net segmentation of tubular structures from live imaging confocal microscopy: Successes and challenges **65**

D Quantifying Topology In Pancreatic Tubular Networks From Live Imaging 3D Microscopy **73**

Chapter 1

Biological Motivation

The purpose of this chapter is to introduce the biological question which this thesis is preparing to answer. For this reason, we first begin by introducing some basic concepts in cell biology. The intention is for the biological notions to be introduced in a way that is accessible to an audience coming from other backgrounds. For a more thorough review, the reader is referred to [1].

1.1 Developmental Biology

An organism begins its development with a single fertilised cell, called zygote, which is the result of the union between an egg and a sperm. That single cell duplicates many times to make us who we are. However, our bodies consist of cells that are made for specific tasks. The process by which a less specialized cell becomes more specialized is known as cell differentiation. Red blood cells, skin cells, and nerve cells are examples of specialized cell types. Cells can be ranked according to how many cell types they can differentiate into. A totipotent cell is one which can differentiate into *all* cell types. A zygote is an example for a totipotent cell. A pluripotent cell can differentiate into *many* cell types. Embryonic stem cells fall into this category. Then there are multipotent cells which can differentiate into *several* cell types like adult stem cells.

What separates two cell types from one another is the kind of proteins molecules they produce. Proteins not only give cells their shape and structure, but it is also proteins which execute nearly all tasks a cell performs. For example, most chemical reactions that cells perform would normally occur at temperatures much higher than those in a cell. Enzymes are specialized proteins which catalyze (speed up) a specific reaction among many possible ones. There are general-purpose proteins which are common among several cell types: structural proteins provide mechanical support to cells; transport proteins carry small molecules or ions; motor molecules generate movement in cells. There are also specialized proteins which are exclusive to a few cell types. For instance, hemoglobin protein is made only in red blood cells.

Thus, what a cell does depends on the kind of proteins it produces. But what determines what proteins a cell must produce? All cells in our body contain the same double-helix strand of DNA. Therefore, what causes two cell types to produce different proteins cannot come from the DNA. Not all parts in our DNA are used for making proteins. Only those segments in DNA which encode the information to create particular proteins are referred to as genes. Each cell expresses (turns on) a combination of genes to make the proteins it needs to accomplish its myriad tasks.

A useful analogy is to compare DNA to a library, both containing information. Just like all cells have the same DNA, every citizen also has access to the same library. However, what makes a person an engineer and another a philosopher is which books in the library they have read; likewise a cell's *fate* is decided by which genes it has expressed.

The process that maps DNA onto protein is executed in two steps: first from DNA to RNA and then from RNA to protein. In other words, RNA acts as a middleman between DNA and protein. In the first step, the double-helix part in DNA which consists the target gene is unwound and one of the strands is used as a template to copy the DNA information of that segment into RNA. The RNA molecule is as a result much shorter than DNA and single-stranded. Similar to DNA, an RNA sequence also consists of four nucleotide molecules. Since both DNA and RNA sequences are in the language of nucleotides, this step is named "transcription". However, the second step, from RNA to protein, is between two different languages. Hence this step is known as "translation". The letters that specify a protein and its 3D shape are the 20 types of amino acids. Since there are 4 letters in RNA language, every triplet in an RNA sequence is translated into one amino acid, with several RNA triplets leading to the same amino acid. In summary, what is meant by gene expression – using more precise vocabulary – is that the nucleotide sequence of that gene is transcribed into the nucleotide sequence of an RNA molecule, which is then translated into the amino acid sequence of a protein.

A cell can regulate the expression of its genes in response to the cues it receives. These cues can be internal or external. For instance, the nonuniform distribution of a protein in a cell at the time of duplication can act as a cue affecting the fate of the daughter cells. External cues can be short distance signals secreted from other cells or even through direct cell-cell contact, or long distance through hormone proteins which are released by cells in the blood stream. External cues could also originate in the physical environment such as pulling and pushing forces (tension), the composition of the surrounding structural proteins (extra cellular matrix), access to oxygen, temperature, etc. A certain cell fate needs a unique combination of signals and mechanical input in order to differentiate. This combination is available only in one specific place referred to as "the niche".

With recent advancements in stem cell technology, a new window for therapeutic applications has been opened. If we understand the mechanisms affecting the differentiation of desired cell types, we can then push a pluripotent stem cell

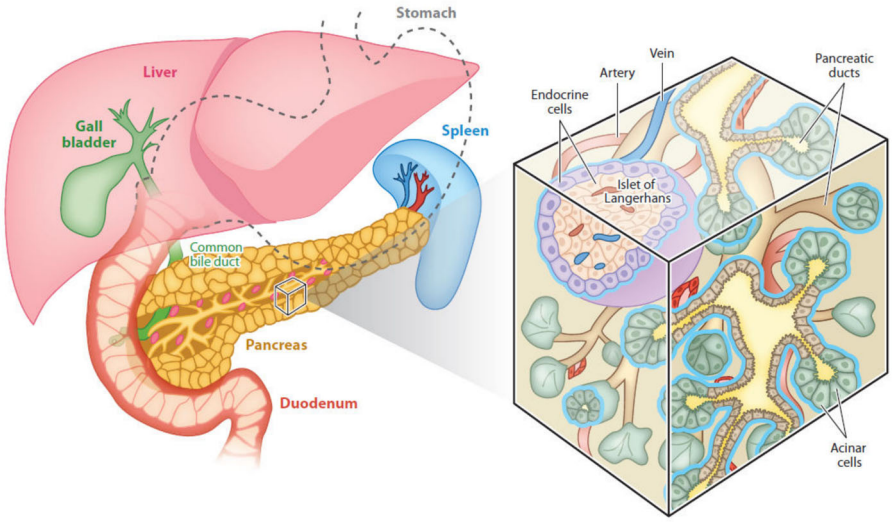


FIGURE 1.1: Image taken from [36]. **Left:** Pancreas location relative to other organs. **Right:** A zoomed in slice showing the enzyme-producing acinar cells situated at the tip of the tubular system, and the islet of Langerhans positioned close to the tubes while being connected to the blood stream to release the produced hormones.

into taking the desired fate in vitro (in test-tube) and use that to regenerate tissues or help patients who lack vital proteins.

1.2 Pancreas Development

With these introductory notes in place, we are now ready to move on to understand how pancreas develops during embryonic stage. For a more complete discussion of pancreas development, we refer the reader to [31, 36].

The pancreas is an organ in the body with exocrine and endocrine functions: the exocrine pancreas secrets a digestive juice for nutrient metabolism, and the endocrine pancreas releases various hormones into the blood stream to regulate the glucose level. Each of these functions involve various cell types and their specialized proteins. The exocrine function is carried out by acinar cells and ductal cells. Acinar cells produce enzymes which catalyze the breakdown of carbohydrates and proteins. The ductal cells in return form a tubular structure that this juice passes through to reach the duodenum. Endocrine cells consist of five cell types, each responsible for the production of a specific hormone. All endocrine cell types live in a small cluster called islet of Langerhans which is located near the tubular system (Figure 1.1). Out of endocrine cell types, only the insulin-producing β-cells are of interest to us in this project.

The pancreas development during embryonic stage is governed by a complex process of signaling and transcription regulation. A better understanding of these events can lead to treatment of pancreatic diseases. Type I diabetes, for instance,

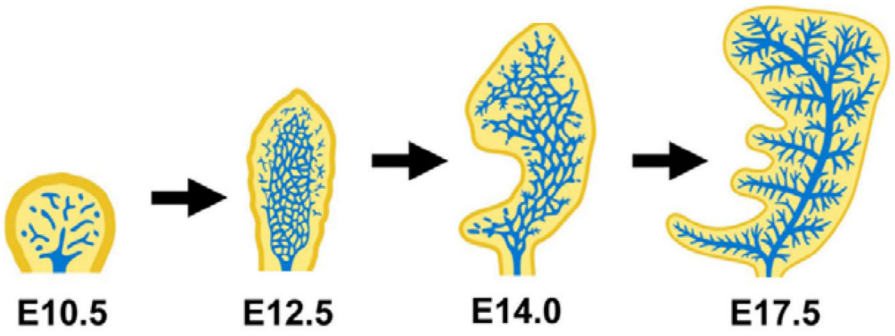


FIGURE 1.2: Image taken from [48]. Mouse pancreas remodeling over several embryonic days.

is a disease where the β-cells are dysfunctional or dead. Transplantation of islets from donors can temporarily cure type I diabetes [35], thus replacement therapy with new β-cells are needed. This can be achieved through engineering human embryonic stem cells to differentiate into the target cell type in vitro, or by reprogramming of pancreatic cell types directly in the body [31].

In mice, pancreas takes its initial form around day E10.5 (embryonic day 10.5). By E12.5, these cells coalesce to make a web-like structure. At E17.5 the pancreas has taken its mature tree-like structure which is more optimal for the transportation of enzymes. During this remodeling from E14 to E17.5, tubular loops eliminate providing the pancreas with its final form [31, 48] (Figure 1.2).

Concurrent to these remodeling events, pancreatic cells are making their cell fate choices. First, they are divided into two groups, namely 'tip' and 'trunk'. Tip cells later become acinar cells. Trunk cells are bipotent meaning that they can either become ductal cells or endocrine cells. Finally, those endocrine cells differentiate into one of the five hormone-producing cell types [36].

It has been suggested in [3, 19] that the remodeling of pancreatic tubes might act as a cue to direct differentiation of trunk cells to either ductal or endocrine cells.

The goal of this PhD project is to provide the tools necessary to investigate this hypothesis computationally. In particular we are interested in extracting topological features which are biologically plausible to affect β-cell differentiation. These topological features could be the opening and closing of loops, their diameter, number of branches etc.

The data set that we analyze is recorded by live-imaging fluorescent 3D microscopy during embryonic stage of mouse pancreas. The pancreas is dissected at E12.5 and grown in incubation chamber for 48 hours. Then they are recorded from E14.5 for 48 hours (assuming the cells are alive) with a frame rate of one every 10 minutes using a Zeiss LSM780 confocal microscope equipped for live imaging. The image resolution is $0.346 \times 0.346 \times 1.0$ μm (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

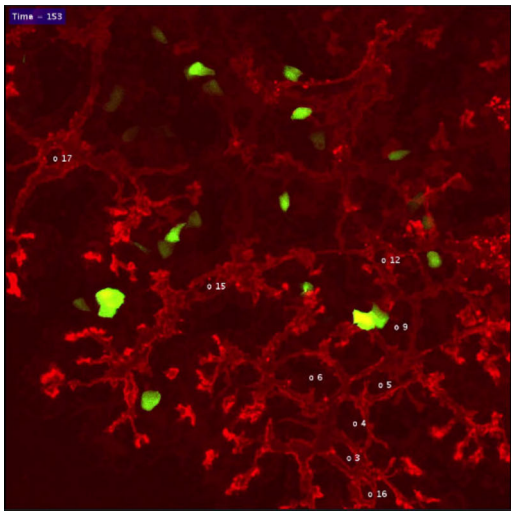


FIGURE 1.3: Maximum intensity projection of a frame of a live imaging movie. Tubular network is shown in red, and insulin-producing β-cells are shown in green. Manually identified loops are given ID numbers (written in white font) to track them over time.

red fluorescent protein (mCherry), which is localized to the that side of the cell facing the tube's inner surface. The green channel captures green fluorescent protein (GFP) to reveal β-cells [29] (Figure 1.3).

Chapter 2

Problem Statement

During embryonic development, the tubular network in the pancreas remodels from a web-like structure to a tree-like structure. Biologists have hypothesized that this morphological change might have an influence on the appearance of nearby β-cells which produce insulin. As a result, careful study of the topological changes that occur in the tubular network are needed to evaluate this hypothesis.

Verifying this biological hypothesis has inspired segmenting the tubular structure from the live imaging data of mice pancreas during embryonic stage. In general, live imaging is challenged by low signal-to-noise ratio (as compared to fixed tissue imaging), due to the fact that cells need to be kept alive, which translates into low laser power and fast scanning speed. This makes segmenting the tubular structure a difficult task even for trained biologists. The challenges raised by the data are discussed at more length in Appendices C and D. An example of the segmentation task is shown in Figure 2.1.

Segmentation requirements. For a viable test of the biological hypothesis, we would need *reliable* and *efficient* segmentations. While manual detection of human experts are in general reliable, they are not efficient. It takes a long time for a human to classify each voxel in the 3D volumetric data. A single 3D image takes about 50 hours to annotate. This makes the manual approach unattainable for a scalable study of several movies, which consist of more that a hundred frames each. Moreover, conventional automated image segmentation methods such as intensity thresholding does not suit this dataset since the fluorescent protein lies only on the surface of the tubes, leading to a hollow segmentation.

On the other hand, deep learning models are very efficient: once the model is trained, it usually takes a few minutes to make predictions for new input. Deep learning models have seen a huge success in a variety of areas such as image classification [44], artificial voice generation [30], and protein folding [18] to name a few. In fact, they have outperformed humans in a multitude of tasks [4, 14]. Thus it seems only natural to employ these successful models to replace the manual detection of tubular structure as well. However, they pose two big challenges:

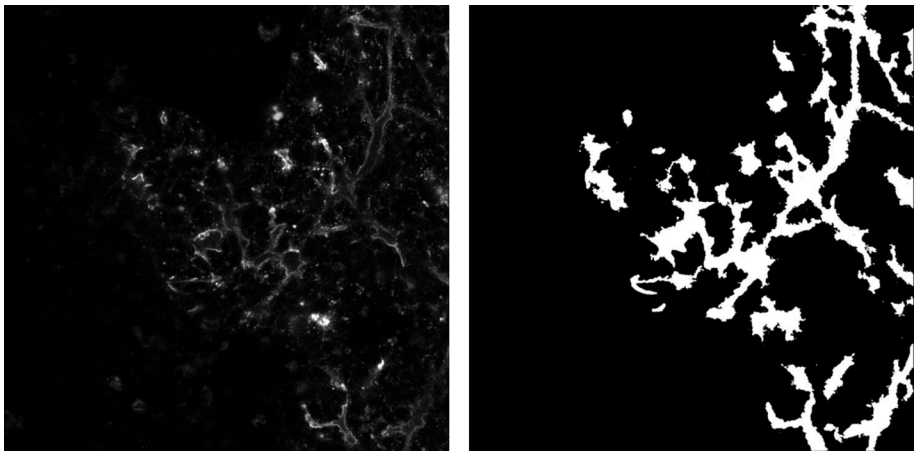


FIGURE 2.1: **Left:** A 2D slice of the input image **Right:** Ground truth segmentation provided by an expert.

1. The high performance of these models is predicated on having large amounts of labeled training data [43].
2. Their predictions are not always reliable [42].

Cost of annotations. Although deep learning models are the state-of-the-art in segmentation tasks, they require large amounts of annotated data to reach their full capacity. This is indeed a serious shortcoming for our application, as not only is it cumbersome to annotate 3D images on a voxel level, but also expert knowledge is costly to attain. We are thus forced to come up with a strategy to overcome this restriction. We experiment with two methods to cope with this problem:

1. Active learning (Section 3.3) which aims to make efficient use of the annotation budget to solve the segmentation problem.
2. Semisupervised segmentation (Appendix D) which uses unannotated data to help with the segmentation task.

Active learning. Rather selecting samples at random to be annotated, active learning selects carefully selects a subset of samples to be labeled. There have been several suggestions in the literature [33, 49] on how to select the most useful samples. One popular line of thought is based on selecting the most uncertain samples [21]. This approach can easily be applied to probabilistic models. In our experiments, a common phenomenon was that models were mostly certain about the inside of the tubes while having a high uncertainty near the boundary of the tubular structure (Figure 2.2). These are also challenging for the human annotator. While this might not be a problem in itself, it inspired us to delve more into the topic of active learning. What hinders using an active learning setting is that it is not clear ahead of time which sampling strategy is going to be better than mere random selection of samples to be annotated [23]. This poses a serious shortcoming to the practical use of active learning as one might actually lose performance by selectively choosing samples to be labeled. This is the inspiration behind the work presented in Appendix A, which investigates

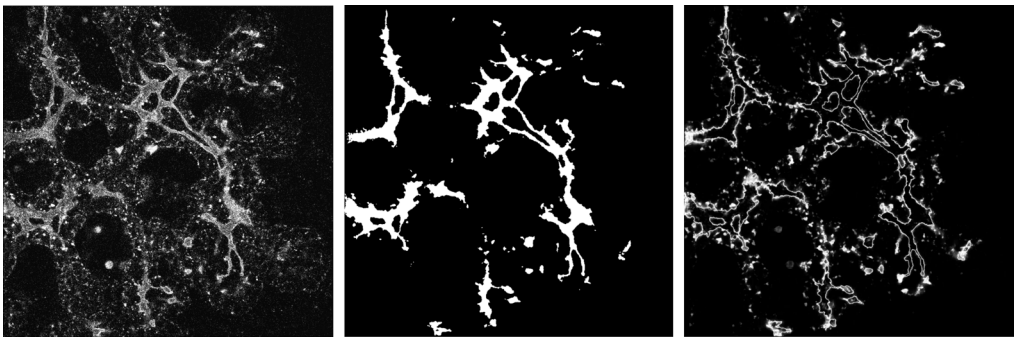


FIGURE 2.2: **Left:** A 2D slice of the input image **Middle:** The corresponding prediction obtained from a deep learning model **Right:** Entropy of the image which shows more uncertain pixel as whiter.

whether we can come up with a data gathering strategy that is guaranteed to outperform random sampling.

Semisupervised segmentation. As our labeled training set is necessarily small, it is desirable to use the diverse unlabeled pool of images to learn more generalizable latent space features compared to the case where only labeled images are available. In Appendix D, we experiment with different ways of incorporating unlabeled data into the training and compare them against a fully supervised framework.

Segmentation uncertainty. Endowing segmentation models with uncertainty estimates can be an effective way to bring a sense of reliability into the predictions. In particular for our application, we would like to have an accurate sense of which segmentations are trustworthy and can be reliably used for biological hypothesis testing. There are several models which quantify segmentation uncertainty [10, 20, 32], but it is not clear what these uncertainties represent. In Appendix B, we study whether these uncertainty estimates correlate with segmentation error. We also investigate whether these uncertainties are beneficial for the purpose of active learning. In Appendix D, we show that for our data, these uncertainty estimates correlate with segmentation difficulty which has been assessed by a laboratory assistant annotating ground truth segmentations.

Topology of the tubular network. Segmentation tasks are usually trained on metrics that ensure pixel-wise accuracy, whereas for our application what is of crucial importance is to correctly predict the topology of the structure. There could be several cases where a few incorrect pixel-wise predictions could disrupt the topology of the tubular network. Moreover, tubular loops play a prominent role in examining the biological hypothesis. This has motivated us to propose a topological score function in Appendix D, which can be broken down to loop- and component subscores. Since it is the changes over time which are important, in Appendix D, loops are tracked over the duration of the movies and the results are compared to annotations by a human expert.

Chapter 3

Machine Learning Background

The purpose of this chapter is to prepare the reader for the theoretical aspects presented in the appendices. Although we have attempted to make the material comprehensible, basic familiarity with probability theory and machine learning might be of use. The examples provided throughout this chapter are intended to provide insight into machine learning problems in a new light. We begin in Section 3.1 by introducing Bayesian inference which assigns probabilities to hypotheses, taking into account the observed data and our prior belief. Section 3.2 uses the probability distributions provided by Bayesian inference to quantify information. In Section 3.3, we discuss how to make use of these concepts to draw samples which are expected to contain the most information about the model parameters.

3.1 Bayesian Inference

It is worth spending some time on the Bayesian definition of probability, as it is the foundation on which Bayesian framework is built upon. Those who are interested in a comparison of schools of thought in probability are encouraged to read [13, 17, 24]. For further reading on Bayesian inference, the reader is directed to [6, 27].

What is probability? The Bayesian definition of probability is the degree of belief for propositions given the collected data. In this view, randomness is not an inherent property of the world, but rather a consequence of our limited knowledge. Even the outcome of a coin toss is deterministic in presence of full knowledge [41]. Therefore, Bayesians reason about non-random unknowns under imperfect knowledge.

The role of Bayesian inference in data modeling is to answer the following two questions given the observed data [25]: 1. Assuming a model is true, which values are more plausible for its parameters? 2. Given a set of models, how plausible are each of them? We refer to the first question as *model fitting* and the second as *model comparison*.

3.1.1 Model Fitting

Bayes' theorem provides a set of tools to update our belief in hypotheses given data. Let us assume that the model $\mathcal{H}_i$, which has parameter vector $\mathbf{w} \in \mathbb{R}^k$, is 'true', meaning it has either generated the data D or it is useful to assume so. Then our posterior belief about the values the parameters $\mathbf{w}$ can take can be expressed as

$$P(\mathbf{w}|D, \mathcal{H}_i) = \frac{P(D|\mathbf{w}, \mathcal{H}_i)P(\mathbf{w}|\mathcal{H}_i)}{P(D|\mathcal{H}_i)}, \tag{3.1}$$

where $P(\mathbf{w}|D, \mathcal{H}_i)$ is the posterior belief of $\mathbf{w}$, $P(D|\mathbf{w}, \mathcal{H}_i)$ is the likelihood, $P(\mathbf{w}|\mathcal{H}_i)$ is the prior belief of $\mathbf{w}$, and $P(D|\mathcal{H}_i)$ is the normalizing constant computed as

$$P(D|\mathcal{H}_i) = \int P(D|\mathbf{w}, \mathcal{H}_i)P(\mathbf{w}|D)d\mathbf{w}. \tag{3.2}$$

The likelihood measures the probability of the data D if we knew $\mathbf{w}$. The choice of prior is subjective, and it quantifies our knowledge about $\mathbf{w}$ prior to seeing the data. If our purpose is to only compare the probabilities of different values for $\mathbf{w}$, then it is not necessary to compute the normalizing constant. However, as we will explain shortly, if we want to compare different models, we need to compute it. This is why the normalizing constant has also been referred to as the evidence of $\mathcal{H}_i$.

In Bayesian inference every statement is expressed in terms of probability and there are no binary decisions about the truthfulness of a particular hypothesis. A hypothesis can only be extremely likely or highly implausible, but not right or wrong. We will now introduce two non-Bayesian procedures used in machine learning to train models (i.e. infer model parameters). Our goal is to see how they are related to Bayesian inference.

Maximum likelihood estimation (MLE). In this approach, we ignore the prior distribution and 'let the data speak for itself'. The parameter which maximizes the likelihood is the selected as the estimated parameter i.e.

$$\hat{\mathbf{w}}_{\text{MLE}} = \arg\max_{\mathbf{w} \in \mathbb{R}^k} P(D|\mathbf{w}, \mathcal{H}_i). \tag{3.3}$$

Maximum a posteriori (MAP) estimate. In this approach, the prior is included in making inference, but only the parameter that maximizes the posterior is taken as the estimated parameter. In other words,

$$\hat{\mathbf{w}}_{\text{MAP}} = \arg\max_{\mathbf{w} \in \mathbb{R}^k} P(\mathbf{w}|D, \mathcal{H}_i). \tag{3.4}$$

Let us now get concrete and see how the introduced methods treat a simple inference problem.

Inference example [27]: Assume a coin has a fixed probability f of coming up heads. We observe n heads in N tosses. What can we infer about the value

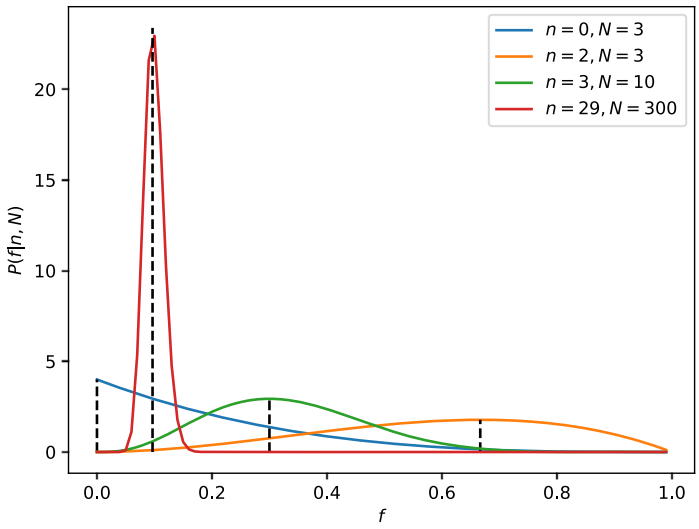


FIGURE 3.1: Posterior distribution of coin bias f given fixed values for the number of heads n and total tosses N. Note that the peaks occur at n/N and when N is large, the posterior is sharply peaked.

of f assuming total ignorance about its prior value?

We assume a uniform prior on f, i.e. $P(f) \equiv 1$. We know the fact that $\int_0^1 x^a(1-x)^b dx = a!b!/(a+b+1)!$ and that $P(n|N,f)$ is a binomial distribution. Therefore, the posterior of f is

$$P(f|n,N) = \frac{P(n|N,f)P(f)}{\int_0^1 P(n|N,f)P(f)df} = \frac{(N+1)!f^n(1-f)^{N-n}}{n!(N-n)!}. \tag{3.5}$$

The posterior distribution of f is shown for four values of n and N in Figure 3.1.

Now, let us review the MLE and MAP answers to this inference problem. Since we have assumed a flat prior over the possible values for f, both MLE and MAP lead to the same results. Both the likelihood and the posterior peak at $\hat{f} = n/N$ and $\hat{f}$ is chosen as the estimated value for the probability of coming up heads for this coin.

From inference to prediction. Now let us ask what is the probability the next toss is a heads? To answer that question, we need to take in account the uncertainty that we have about f, so we should compute $\mathbb{E}_{P(f|n,N)}[P(t_{N+1} = \text{heads}|N,n,f)]$, in which $\mathbb{E}[\cdot]$ denotes the expected value. Thus, we can write

$$P(t_{N+1} = \text{heads}|n,N) = \int_0^1 P(f|n,N)P(t_{N+1} = \text{heads}|f,n,N)df. \tag{3.6}$$

If we know the probability of coming up heads f, then we do not need the observations to predict the next toss, in other words $P(t_{N+1} = \text{heads}|f, n, N) = P(t_{N+1} = \text{heads}|f) = f$. Using this and the posterior (3.5), we can compute (3.6) as

$$P(t_{N+1} = \text{heads}|n, N) = \frac{n+1}{N+2}. \tag{3.7}$$

This is the correct and exact way to make predictions when using Bayesian inference. The MLE and MAP have already inferred $\hat{f} = n/N$, so their prediction is $P(t_{N+1} = \text{heads}|\hat{f}) = n/N$, which is very different from the Bayesian prediction when the number of observations are low. Having said that, when data points are numerous the MAP estimate might be a good approximation of the posterior, since in such cases the posterior is usually sharply peaked and can be approximated by a single value at its maximum.

3.1.2 Model Comparison

In this level of inference, we would like to compute the posterior belief over models (to within a normalising constant) i.e.

$$P(\mathcal{H}_i|D) \propto P(D|\mathcal{H}_i)P(\mathcal{H}_i). \tag{3.8}$$

If we set a flat prior over all models, i.e. if we do not have a reason to believe one of the models is more plausible than the others, then model selection only depends on the evidence $P(D|\mathcal{H}_i)$.

Occam's razor states that among models which fit the data, those which are less complex (have fewer parameters) are preferred. A closer look at the evidence Equation (3.2) reveals why Occam's razor is embodied in Bayesian model comparison [25]. Under sufficient data, the posterior distribution over the parameters are usually strongly peaked (Figure 3.2). If so, we can approximate the integral in Equation (3.2) by the height at its peak times the width as

$$P(D|\mathcal{H}_i) \simeq P(D|\mathbf{w}_{\text{MAP}}, \mathcal{H}_i)P(\mathbf{w}_{\text{MAP}}|\mathcal{H}_i)\Delta\mathbf{w}_{\text{posterior}}. \tag{3.9}$$

If we assume a flat prior with width $\Delta\mathbf{w}_{\text{prior}}$, i.e. $P(\mathbf{w}|\mathcal{H}_i) = 1/\Delta\mathbf{w}_{\text{prior}}$, then we can rewrite Equation (3.9) as

$$P(D|\mathcal{H}_i) \simeq P(D|\mathbf{w}_{\text{MAP}}, \mathcal{H}_i)\frac{\Delta\mathbf{w}_{\text{posterior}}}{\Delta\mathbf{w}_{\text{prior}}}, \tag{3.10}$$

where $\Delta\mathbf{w}_{\text{posterior}}/\Delta\mathbf{w}_{\text{prior}}$ is named "Occam's factor" [13]. A more complex model would have a higher $\Delta\mathbf{w}_{\text{prior}}$ and consequently a lower evidence meaning it would be ranked lower compared to a simpler model.

Now that we have given an introduction to Bayesian inference, let us go back the data modeling process shown in Figure 3.3. Other than model fitting and model comparison, one can use the information obtained from the evidence to decide if new models are needed to be considered. For instance, if the posterior distribution over the hypotheses is not sharply peaked, then we might consider

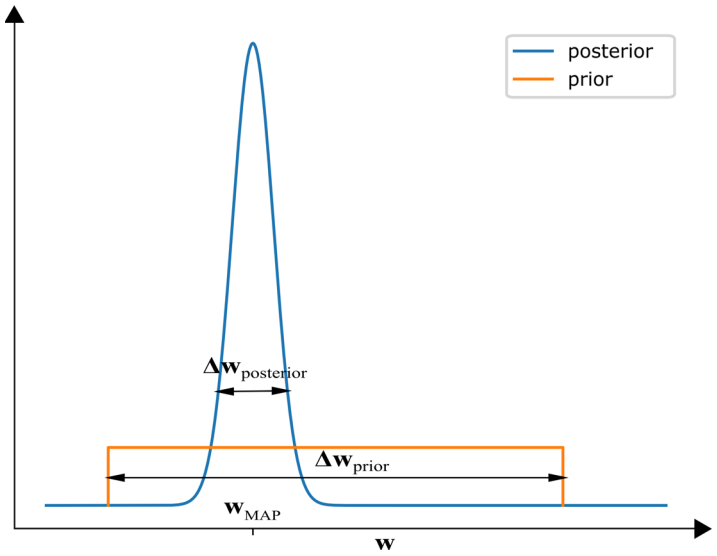


FIGURE 3.2: Image taken from [25]. The reason why Bayesian model comparison includes Occam's razor: The ratio between the width of the posterior ($\Delta \mathbf{w}_{\text{posterior}}$) to the width of the prior ($\Delta \mathbf{w}_{\text{prior}}$) is known as the Occam's factor which punishes model complexity.

rethinking our models and propose a new one and repeat the comparison. Be that as it may, Bayesian inference does not inform us on how to come up with new models, it can only compare their plausibility relative to other models given the data and our assumptions.

Intractability of the integrals. We need to compute two integrals in Bayesian inference:

1. When converting inference to prediction, we need to compute the expected prediction over the posterior distribution of parameters, e.g. in Equation (3.6).
2. When comparing models we need to compute the integral in Equation (3.2) to calculate the evidence.

The reason for this intractability is that for many models the posterior would be a complex distribution. There are three approaches to deal with complex distributions:

1. Laplace's method, explained in detail in Appendix A, approximates the posterior by a Gaussian distribution
2. Monte Carlo methods [11, 28, 45] use random numbers to draw independent samples from nasty distributions which can then be used to compute expectations on those distributions and also calculate the normalizing constant.

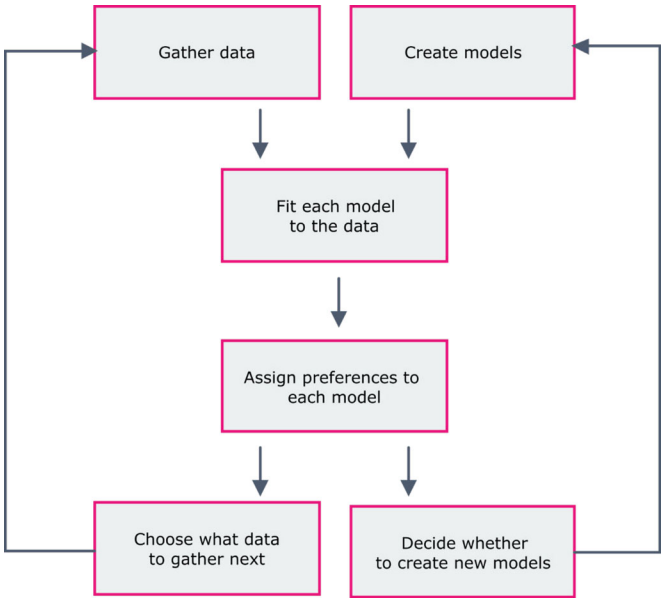


FIGURE 3.3: Image adapted from [25]. Data modeling process: Bayesian inference provides straightforward answers to model fitting and model comparison. We can then use these answers to decide if we need new models or what data to gather next.

3. Variational methods [5, 12, 15] approximate nasty distributions by well-behaved parameterized distributions and then minimize the Kullback–Leibler divergence to find the parameters.

Despite these estimations, it is still computationally expensive to use Bayesian approach for model comparison. For this reason, we use a validation set to compare models throughout this thesis.

3.2 Information Theory

Bayesian inference can tell us what data to gather next, also known as active learning. To do this, we need to come up with a measure of information and then select the data point that is maximally informative for either model fitting or model comparison [26].

In this section, we begin by introducing how to measure information for discrete random variables. These definitions are pooled in the 'weighing problem' to give some profound insight into the active learning problem. We then introduce what properties hold for continuous random variables. For further reading on the topic of information theory, we highly recommend [27].

What is information? How can one measure it? In his seminal paper [34], Shannon answered both questions.

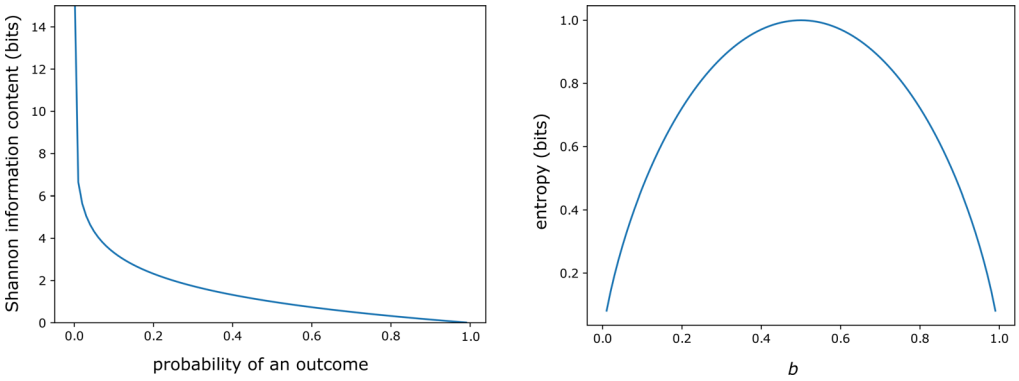


FIGURE 3.4: **Left:** Shannon information content of an outcome. Rare outcomes contain more information than more frequent ones. **Right:** Entropy of a Bernoulli random variable with bias b. The entropy is highest when $b = 0.5$.

The Shannon information content of an outcome x is defined as

$$h(x) = \log_2 \frac{1}{P(x)}, \tag{3.11}$$

which is measured in bits since we are using $\log_2$, but other bases for the logarithm are also possible. Figure 3.4 (Left) gives an the intuition behind this definition, which is that rare outcomes are more informative than frequent ones. Also if an outcome is certain ($P(x) = 1$), the information content for that outcome is zero. The information content is non-negative. Since we do not know the outcome of experiments before we conduct them, it is useful to define the expected information content.

The entropy of a discrete random variable X is then defined as the expected information content i.e.

$$H(X) = \sum_x P(x) \log_2 \frac{1}{P(x)}, \tag{3.12}$$

where if $P(x) = 0$ then $P(x) \log_2[1/P(x)] = 0$. Note that like information content of an outcome, the entropy is also non-negative.

As an example, Figure 3.4 (Right) depicts the entropy of a Bernoulli random variable X with bias b, i.e. $P(X = 1) = b$ and $P(X = 0) = 1 - b$, which is

$$H(X) = b \log_2 \frac{1}{b} + (1 - b) \log_2 \frac{1}{1 - b}. \tag{3.13}$$

We can see that if the outcome is already known, i.e. $b = 0$ or $b = 1$, the entropy is zero. Also, the entropy is highest when the outcome is most uncertain i.e. $b = 0.5$.

The number of possible outcomes in a Bernoulli random variable is two and the entropy is highest when both of these outcomes are equally probable. The

general rule is that given a set of all possible outcome $\mathcal{A}_x$

$$H(X) \leq \log_2(|\mathcal{A}_x|), \tag{3.14}$$

with equality if and only if $P(x) = 1/|\mathcal{A}_x|$ for all x, in other words if $P(x)$ is uniform. Here, $|\cdot|$ when applied on sets denotes the number of elements in that set. For this reason, the entropy measures how *random* or *uncertain* or *unpredictable* a random variable is.

In many scenarios, we are studying more than one random variable. We will now introduce the joint and conditional entropies.

The joint entropy of X, Y is

$$H(X,Y) = \sum_{x,y} P(x,y) \log_2 \frac{1}{P(x,y)}. \tag{3.15}$$

If two random variables are independent i.e. $P(x,y) = P(x)P(y)$, then their joint entropy is additive i.e. $H(X,Y) = H(X) + H(Y)$. This is in accordance to the intuition we have about information.

The conditional entropy of X given $y = y_0$ is the entropy of the probability distribution $P(X|y = y_0)$, i.e.

$$H(X|y = y_0) = \sum_{x} P(x|y = y_0) \log_2 \frac{1}{P(x|y = y_0)}. \tag{3.16}$$

The conditional entropy of X given Y is the average, over y, of the conditional entropy of X given y.

$$H(X|Y) = \sum_{y} P(y) \left[\sum_{x} P(x|y) \log_2 \frac{1}{P(x|y)} \right] = \sum_{x,y} P(x,y) \log_2 \frac{1}{P(x|y)} \tag{3.17}$$

The chain rule for entropy:

$$H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y). \tag{3.18}$$

If X and Y were independent then $H(Y|X) = H(Y)$ and $H(X|Y) = H(X)$, and we would have the additive property of entropies of independent random variables.

The mutual information between X and Y is

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X). \tag{3.19}$$

The mutual information is symmetric, i.e. $I(X;Y) = I(Y;X)$, and $I(X;Y) \geq 0$. The mutual information between two independent random variables is zero. Intuitively, the mutual information measures the average amount of information that x conveys about y. In other terms, how much on average the uncertainty of x is reduced once we learn y.

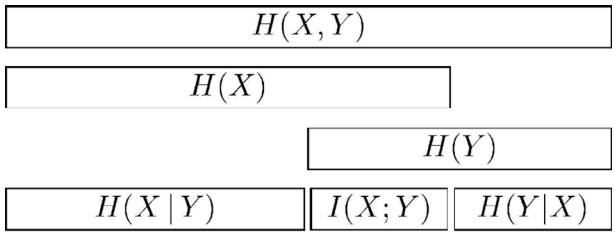


FIGURE 3.5: Image taken from [27]. The relationship between joint entropy, marginal entropy, conditional entropy, and mutual information.

Figure 3.5 summarizes all the formulas about the entropies of two dependent random variables.

The weighing problem. Imagine you are given 12 balls, all of which except for one weigh the same. You are given a two-pan balance which can measure whether the left side is heavier, the right side is heavier, or they weigh the same. How would you use the scale to find out which of the 12 balls is the odd one *and* whether it is heavier or lighter while using the scale as few times as possible?

We know we have 24 hypotheses (12 balls each being heavy or light). There are three states the scale can be in, so it is impossible to distinguish the odd ball after 2 uses of the scale since $3^2 < 24$, but we might be able to do so in 3 uses or more.

Let us denote the input of the experiment for the first usage by X, showing which balls are weighed against which. The output of the scale for the first usage is denoted by Y. The hypotheses about which ball is the odd one is shown by W. There are 24 candidates and initially, without having used the scale at all, they are all equally probable, i.e. $P(w_i) = 1/24$ and $H(W) = \log_2 24$. A greedy strategy would be to conduct experiments which are maximally informative at every step[1]. Such a strategy would choose x such that the expected uncertainty in w is reduced the most. The reason for the word 'expected' in the last sentence is that we do not know the outcome of the weighing. The mutual information $I(W;Y|X)$ is the quantity we would like to maximize. We can compute the mutual information in two ways. We begin with $I(W;Y|X) = H(W|X) - H(W|Y,X)$. Knowing x alone would not reveal anything about w, so $H(W|X) = H(W) = \log_2 24$. For the first usage, there are 6 options. We can weigh 6 balls against 6 and leave no ball on the ground, or we can weigh 5 balls against 5 and leave 2 balls on the ground etc. We denote these experiments by 6v6, ..., 1v1. For example, if we use 6v6, we know the scale will not balance, and by symmetry it is equally likely that either side is heavier. In any case, this reveals that half of the balls could not be heavy and the other half could not be light and the probability for

[1]This strategy could in principle give rise to solutions which are not globally optimal, but that does not occur in this example.

the remaining 12 hypotheses would be equally likely. As a result,

$$H(W|Y, X = 6\text{v}6) = \sum_{y \in Y} P(y|X = 6\text{v}6) \sum_{w \in W} P(w|y, X = 6\text{v}6) \log_2(\frac{1}{P(w|y, X = 6\text{v}6)}) = \log_2 12.$$

Thus $I(W; Y|X) = \log_2(24/12) = 1$ bit. Intuitively, 1 bit of information is the same as asking a yes/no question and it removes half of the hypotheses. It is in general easier to compute $I(W; Y|X) = H(Y|X) - H(Y|W, X)$. The reason being that knowing which ball is the odd one and if it is heaver or lighter, also tells us the outcome of the scale, so $H(Y|W, X) = 0$ regardless of what x is. Consequently, $I(W; Y|X) = H(Y|X)$. For instance when $X = 5\text{v}5$, it will balance if the odd ball is one of the two balls on the ground (2/12 chance), and the rest of the probability is split equally on either side, so $H(Y|X = 5\text{v}5) = (2/12) \log_2(12/2) + (5/12) \log_2(12/5) + (5/12) \log_2(12/5) = 1.48$. The following is the information gain of every possible weighing:

$$\begin{aligned} &I(W; Y|X = 6\text{v}6) = 1, && I(W; Y|X = 3\text{v}3) = 1.5 \\ &I(W; Y|X = 5\text{v}5) = 1.48, && I(W; Y|X = 2\text{v}2) = 1.25 \\ &I(W; Y|X = 4\text{v}4) = 1.58, && I(W; Y|X = 1\text{v}1) = 0.82 \end{aligned}$$

Therefore, for the first usage $X = 4\text{v}4$ is optimal according to mutual information. We will not go into the next steps as it might be harder to explain due to the branching of outcomes without having a proper labeling of balls. However, a full optimal solution with 3 uses of the scale can be found in [27].

The main goal of presenting this example here is to shed some light on how input and output of an experiment can reduce the uncertainty about a hypothesis, and why selecting an input which maximizes the mutual information between the output and the hypotheses could be a profitable rule of thumb. Active learning is also known as the optimal experimental design. The weighing problem is a tangible example of active learning, where we want to infer the true hypothesis using a minimum amount of data. In the next section, we will explore active learning using the same language of inference and information in the context of machine learning.

Since the parameters in machine learning models are continuous we will need to introduce the continuous entropy.

The entropy of a continuous random variable X is defined as

$$S(X) = \int_{-\infty}^{+\infty} P(x) \log_2 \frac{1}{P(x)} dx. \tag{3.20}$$

At first glance, continuous entropy seems like the natural extension of the discrete entropy that Shannon suggested, replacing the sum with an integral. However, let us take a closer look at the continuous entropy by partitioning the continuous random variable X into infinitely many bins (each denoted by x_i) of width Δ. We now define the quantized random variable X^Δ by $X^\Delta = x_i$

if $i\Delta <= X < (i+1)\Delta$. Thus we have $P(X^\Delta = x_i) = P(x_i)\Delta$. We can now compute the entropy of the the discrete random variable X^Δ as $\Delta \to 0$ by

$$\begin{aligned} H(X^\Delta) &= \lim_{\Delta\to 0} \sum_{i=-\infty}^{+\infty} P(x_i)\Delta \log_2 \frac{1}{P(x_i)\Delta} \\ &= \lim_{\Delta\to 0} \sum_{i=-\infty}^{+\infty} P(x_i)\Delta \log_2 \frac{1}{P(x_i)} + \lim_{\Delta\to 0} \sum_{i=-\infty}^{+\infty} P(x_i)\Delta \log_2 \frac{1}{\Delta}. \end{aligned}$$

By Riemann sum we have

$$\lim_{\Delta\to 0} \sum_{i=-\infty}^{+\infty} P(x_i)\Delta \log_2 \frac{1}{P(x_i)} = \int_{-\infty}^{+\infty} P(x) \log_2 \frac{1}{P(x)} dx = S(X)$$

and

$$\lim_{\Delta\to 0} \sum_{i=-\infty}^{+\infty} P(x_i)\Delta = \int_{-\infty}^{+\infty} P(x) dx = 1.$$

As a result, we can write the relation between continuous entropy and discrete entropy as

$$H(X^\Delta) \to S(X) + \log_2 \frac{1}{\Delta}, \quad \text{as } \Delta \to 0. \tag{3.21}$$

The entropy of the discrete distribution diverges as the quantization width gets smaller and smaller. Also, discrete entropy is dependent on quantization width Δ and is increased by 1 bit for every halving of Δ. This makes the continuous entropy to be scale dependent and not be of any use on its own [16].

For instance, the continuous entropy of the uniform random variable $x \sim \mathcal{U}(a, b)$ is $S(X) = \log_2(b-a)$. For $\mathcal{U}(0, 1/2)$, S(X) = -1. The continuous entropy can be negative, and thus cannot be a measure of information by itself. However, all is not lost and the mutual information between two continuous random variables is well-behaved.

The mutual information between two continuous random variables X and Y is

$$I(X;Y) = S(X) - S(X|Y) = S(Y) - S(Y|X), \tag{3.22}$$

The mutual information for continuous random variables has all the properties of its discrete counterpart [9] since

$$I(X^\Delta;Y^\Delta) = H(X^\Delta) - H(X^\Delta|Y^\Delta) = S(X) + \log_2 \frac{1}{\Delta} - S(X|Y) - \log_2 \frac{1}{\Delta} = I(X;Y). \tag{3.23}$$

Now that we have defined the mutual information between two continuous random variables, we are ready to move onto the active learning.

3.3 Active Learning

There are numbers of applications in which obtaining large amounts of data is straightforward. Supervised machine learning models, however, rely on having *labeled* data to train their parameters. These annotations are costly and cumbersome to attain, as they require human knowledge. Active learning poses the following question: If we had a fixed budget for annotation by a human, could we spend them wisely and select samples to be annotated which improve model performance more than a random selection? This careful selection of samples is based on the machine learning model output. However, there have been quite a few suggestions as to how to select the most useful samples. For an introduction to various methods, the reader is directed to [33, 49].

In this section, we will draw on the intuition introduced in the weighing problem of Section 3.2 to select informative samples to fit model parameters. Our goal is to make the connection between mutual information and the change in entropy proposed in Appendix A . To keep our focus on the big picture, our formulation in this section will be model agnostic. The derivation for a logistic regression model has been presented in Appendix A. In Appendix B, we investigated an active learning procedure where samples with the highest uncertainty were selected. Active learning did not work in this setting and this section will also shed some light on the potential reason behind it.

We adopt the mutual information framework introduced in Section 3.2 for active learning. Assume we have gathered N pairs of input and output as $\mathcal{D}_N = \{x_n, y_n\}_{n=1}^N$. Just as the weighing problem, we denote the next input, its corresponding output, and model parameters with X_{N+1}, Y_{N+1}, and W respectively. We would like to choose x_{N+1} such that its expected corresponding output y_{N+1} reveals the most information about the model parameters. In other words, we want to maximize the mutual information between model parameters W and the obtained label Y_{N+1} given the chosen input X_{N+1} and observed data $\mathcal{D}_N$, i.e.

$$x_{N+1} = \arg\max_{x \in \mathcal{X}} I(W; Y_{N+1} | X_{N+1} = x, \mathcal{D}_N), \tag{3.24}$$

where $I(\cdot;\cdot)$ denotes the mutual information between two random variables and $\mathcal{X}$ is the set of unlabeled data we can choose from. Using Equation (3.22) one can write

$$I(W; Y_{N+1} | X_{N+1}, \mathcal{D}_N) = S(W | X_{N+1}, \mathcal{D}_\mathcal{N}) - S(W | Y_{N+1}, X_{N+1}, \mathcal{D}_N). \tag{3.25}$$

Since knowing x_{N+1} alone does not provide information about $\mathbf{w}$ and $\mathcal{D}_{N+1} \equiv \mathcal{D}_\mathcal{N} \cup \{x_{N+1}, y_{N+1}\}$, we can rewrite the above equation as

$$I(W; Y_{N+1} | X_{N+1}, \mathcal{D}_N) = S(W | \mathcal{D}_N) - S(W | \mathcal{D}_{N+1}). \tag{3.26}$$

We can now rewrite Equation (3.24) as

$$x_{N+1} = \arg\max_{x \in \mathcal{X}} \left[S(W | \mathcal{D}_\mathcal{N}) - S(W | \mathcal{D}_{N+1}) \right], \tag{3.27}$$

which is the criterion proposed in Appendix A. Since the goal of data gathering is to find which parameters are more plausible, Equation (3.27) makes intuitive sense to select the sample which reduces the uncertainty over parameters the most, or in other words makes the posterior over parameters as sharply peaked as possible.

We now need to compute the entropy

$$S(W|\mathcal{D}_N) = \int P(\mathbf{w}|\mathcal{D}_N) \log_2 \frac{1}{P(\mathbf{w}|\mathcal{D}_N)} d\mathbf{w}, \tag{3.28}$$

which could be intractable for complex posterior distributions.

Now let us take the alternative approach to computing mutual information–using its symmetry property– to see if we end up in a better position. We have[2]

$$I(W; Y_{N+1}|X_{N+1}, \mathcal{D}_N) = H(Y_{N+1}|X_{N+1}, \mathcal{D}_N) - H(Y_{N+1}|W, X_{N+1}), \tag{3.29}$$

since $H(Y_{N+1}|W, X_{N+1}, \mathcal{D}_N) = H(Y_{N+1}|W, X_{N+1})$.

If the model prediction is a deterministic function of input and parameters, i.e. $f(x; \mathbf{w}) = P(y = 1|x, \mathbf{w})$, then $H(Y_{N+1}|W, X_{N+1}) = 0$.

To compute $H(Y_{N+1}|X_{N+1}, \mathcal{D}_N)$, we need to calculate the following integral:

$$P(y = 1|x, \mathcal{D}_N) = \int P(y = 1|x, \mathbf{w}) P(\mathbf{w}|\mathcal{D}_N) d\mathbf{w}. \tag{3.30}$$

This integral is also intractable for complex posterior distributions.

Common approaches to evaluating intractable integrals were discussed in Section 3.1.

We hypothesize that the failure of the active learning experiments conducted in Appendix B might be due to the following reasons:

1. Use of the MAP estimation in some of the models
2. Presence of label uncertainty

MAP estimation in active learning. One might use the MAP estimate to approximate the integral in Equation (3.30) as

$$P(y = 1|x, \mathcal{D}_N) \simeq P(y = 1|x, \mathbf{w}_{\text{MAP}}) = f(x; \mathbf{w}_{\text{MAP}}). \tag{3.31}$$

However, active learning is especially useful when the number of observed samples N is small, thus the posterior distribution would not be sharply peaked (see Figure 3.1) and the MAP estimate is inaccurate. As a result, active learning approaches such as [21, 46] which maximise the entropy of the model prediction

[2]We are assuming we are dealing with a binary classification task, meaning Y_{N+1} is a discrete random variable, hence the use of discrete entropy. Regression tasks are similar, only the continuous entropy would be used.

at $\mathbf{w}_{\text{MAP}}$ or equivalently take samples close to the decision boundary at $\mathbf{w}_{\text{MAP}}$ are most likely not going to work.

Label uncertainty in active learning. Thus far in this thesis, we have assumed that the label we receive from the human expert is absolutely correct, while in practice the annotator is often uncertain about his/her annotation. To account for this label uncertainty, we can modify Y to be a continuous random variable between 0 and 1, which measures how certain the annotator is that the input belongs to the positive class. As it might be difficult for annotators to gauge their confidence, we can ask several annotators for their binary opinions and average the outcomes (assuming they are equally capable). Making the random variable Y continuous would mean the entropies in Equation (3.29) should be replaced with continuous entropies, which might be difficult to compute. Another aspect which would be affected by this change is the likelihood function, which would be related to continuous cross-entropy rather than binary cross-entropy. All these changes require careful study and reformulation, which can be considered for future work.

Chapter 4

Summary and Future Work

This chapter is devoted to summarizing the work presented in the appendices, and to discussing possible future directions for the research carried out during this PhD project. The summary is kept brief, as more detailed summaries will be given in each appendix.

This thesis presents the problem of segmenting the tubular structure in the pancreas from confocal microscopy live imaging data. In Chapter 1, we provided the basic biological background necessary to understand the biological motivation behind this thesis. Chapter 2 outlined the challenges that were encountered in this project and how each appendix contributes to providing solutions to these problems. A machine learning background to facilitate the understanding of appendices is given in Chapter 3, consisting of Bayesian inference, information theory, and active learning.

In Appendix A, we look into the active learning problem from an information theory perspective. We propose a criterion which selects samples that provide the most information gain on model parameters. For a logistic regression model, we show that the criterion has a nice intuition. At the start when the number of samples are low, it selects samples from the edge of the data space. When more and more labels are acquired, samples which are close to the decision boundary are preferred. Therefore, it provides an adaptive trade-off between exploration and exploitation.

The work in Appendix A comes with two main drawbacks:

1. The derivation assumes that the model has given rise to the data.
2. It is difficult to compute the entropy for complicated models.

Unfortunately, Bayesian inference does not include 'model criticism' [6], and all inferences are performed under the assumption the the hypothesis space is correct. One solution could be to choose more complicated models and have suitable priors to rectify overfitting. Adopting this approach, however, leads us to the second weakness. The posterior distribution for complex models is usually multimodal and it is intractable to compute the entropy. A future direction of work would be to use Markov Chain Monte Carlo (MCMC) [28] methods to compute the entropy. While traditional MCMC methods were not scalable for

high dimensional problems, more recent work in nested sampling [37, 40] and Galilean Monte Carlo [38, 39] look promising. Another line of thought would be to apply the information gain criterion on Gaussian processes. Gaussian processes are non-parametric models, so we can select informative samples for the hyperparameters in the covariance function—which are not more than a handful and thus do not pose the scalability challenge.

Appendix B compares four established probabilistic segmentation models for their use case in segmentation error and active learning. We find that that pixel-level uncertainty estimates are indicative of segmentation errors, but are not sensitive to uncertainty modeling. As for the active learning, images with the highest sum of pixel uncertainty were labeled which led to no consistent benefit for any of the models. As discussed in Section 3.3, this approach assumes the posterior is sharply peaked which seems implausible under limited data regime. One interesting question is how one should measure uncertainty of an image for a segmentation task, especially for models such as MC dropout [10] and probabilistic U-net [20] which provide plausible segmentation images rather than pixel-wise uncertainties.

In Appendix C, we delve into the problems we encountered in employing a U-net architecture to the segmentation task. Since our training set is small, it does not cover the diversity that exists among the large number of collected movies. Several movies have intensity distributions which are far away from what is present in the training movies. We tackle this issue in part by preprocessing each image by its own statistics. This improves the predictions for the movies with low signal-to-noise ratio which were prior to this preprocessing predicted as mostly background class.

Two of the issues raised in Appendix C were addressed in Appendix D. One is to use the large pool of unlabeled data to learn better features for the segmentation task. We employed the joint optimization of an autoencoder and a U-net which proved to have a positive impact on the segmentation performance compared to a fully supervised U-net trained only on labeled data. The other issue was that our application required accurate extraction of topology while most deep learning models are trained for pixel-wise performance. Using the GEL skeletonization algorithm [7], we convert the segmentations to graphs. We, then, derive a topological score function, which measures the topological similarity (in terms of loops and components) between a ground-truth graph and a predicted graph. This score function was used for model selection and hyperparameter tuning. Tubular loops are particularly interesting to the biologists. As a result, the predicted loops were tracked over time to study their progression. We also made use of this tracking to get rid of loop trajectories which appeared only briefly.

The project is at a stage where biologically interesting features can be readily extracted and studied over time to have a better understanding of the morphological events that occur during the remodeling of the tubular network. This also means that the tools required to analyse the biological hypothesis are ready to be utilized.

Bibliography

[1] Alberts, B., Bray, D., Hopkin, K., Johnson, A.D., Lewis, J., Raff, M., Roberts, K., Walter, P.: Essential cell biology. Garland Science (2015)

[2] Armato III, S.G., McLennan, G., Bidaut, L., McNitt-Gray, M.F., Meyer, C.R., Reeves, A.P., Zhao, B., Aberle, D.R., Henschke, C.I., Hoffman, E.A., et al.: The lung image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans. Medical physics **38**(2), 915–931 (2011)

[3] Bankaitis, E.D., Bechard, M.E., Wright, C.V.: Feedback control of growth, differentiation, and morphogenesis of pancreatic endocrine progenitors in an epithelial plexus niche. Genes & development **29**(20), 2203–2216 (2015)

[4] Berner, C., Brockman, G., Chan, B., Cheung, V., Dębiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., et al.: Dota 2 with large scale deep reinforcement learning. arXiv preprint arXiv:1912.06680 (2019)

[5] Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: A review for statisticians. Journal of the American statistical Association **112**(518), 859–877 (2017)

[6] Box, G.E., Tiao, G.C.: Bayesian inference in statistical analysis, vol. 40. John Wiley & Sons (2011)

[7] Bærentzen, A., et al.: The GEL library. `http://www2.compute.dtu.dk/projects/GEL/` (2020)

[8] Clark, K., Vendt, B., Smith, K., Freymann, J., Kirby, J., Koppel, P., Moore, S., Phillips, S., Maffitt, D., Pringle, M., et al.: The cancer imaging archive (tcia): maintaining and operating a public information repository. Journal of digital imaging **26**(6), 1045–1057 (2013)

[9] Cover Thomas, M., Thomas Joy, A.: Elements of information theory. New York: Wiley **3**, 37–38 (1991)

[10] Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: international conference on machine learning. pp. 1050–1059. PMLR (2016)

[11] Gilks, W.R., Richardson, S., Spiegelhalter, D.: Markov chain Monte Carlo in practice. CRC press (1995)

[12] Graves, A.: Practical variational inference for neural networks. Advances in neural information processing systems **24** (2011)

[13] Gull, S.F.: Bayesian inductive inference and maximum entropy. In: Maximum-entropy and Bayesian methods in science and engineering, pp. 53–74. Springer (1988)

[14] Hekler, A., Utikal, J.S., Enk, A.H., Solass, W., Schmitt, M., Klode, J., Schadendorf, D., Sondermann, W., Franklin, C., Bestvater, F., et al.: Deep learning outperformed 11 pathologists in the classification of histopathological melanoma images. European Journal of Cancer **118**, 91–96 (2019)

[15] Hoffman, M.D., Blei, D.M., Wang, C., Paisley, J.: Stochastic variational inference. Journal of Machine Learning Research **14**(5) (2013)

[16] Jaynes, E.T.: Information theory and statistical mechanics. Physical review **106**(4), 620 (1957)

[17] Jaynes, E.T., Kempthorne, O.: Confidence intervals vs bayesian intervals. In: Foundations of probability theory, statistical inference, and statistical theories of science, pp. 175–257. Springer (1976)

[18] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al.: Highly accurate protein structure prediction with alphafold. Nature **596**(7873), 583–589 (2021)

[19] Kesavan, G., Sand, F.W., Greiner, T.U., Johansson, J.K., Kobberup, S., Wu, X., Brakebusch, C., Semb, H.: Cdc42-mediated tubulogenesis controls cell specification. Cell **139**(4), 791–801 (2009)

[20] Kohl, S.A., Romera-Paredes, B., Meyer, C., De Fauw, J., Ledsam, J.R., Maier-Hein, K.H., Eslami, S., Rezende, D.J., Ronneberger, O.: A probabilistic u-net for segmentation of ambiguous images. arXiv preprint arXiv:1806.05034 (2018)

[21] Lewis, D.D., Gale, W.A.: A sequential algorithm for training text classifiers. In: SIGIR'94. pp. 3–12. Springer (1994)

[22] Li, Y., Shen, L.: Skin lesion analysis towards melanoma detection using deep learning network. Sensors **18**(2), 556 (2018)

[23] Loog, M., Yang, Y.: An empirical investigation into the inconsistency of sequential active learning. In: 2016 23rd international conference on pattern recognition (ICPR). pp. 210–215. IEEE (2016)

[24] Loredo, T.J.: From laplace to supernova sn 1987a: Bayesian inference in astrophysics. In: Maximum entropy and Bayesian methods, pp. 81–142. Springer (1990)

[25] MacKay, D.J.: Bayesian interpolation. Neural computation **4**(3), 415–447 (1992)

[26] MacKay, D.J.: Information-based objective functions for active data selection. Neural computation **4**(4), 590–604 (1992)

[27] MacKay, D.J., Mac Kay, D.J.: Information theory, inference and learning algorithms. Cambridge university press (2003)

[28] Neal, R.M.: Probabilistic inference using Markov chain Monte Carlo methods. Department of Computer Science, University of Toronto Toronto, ON, Canada (1993)

[29] Nyeng, P., Heilmann, S., Löf-Öhlin, Z.M., Pettersson, N.F., Hermann, F.M., Reynolds, A.B., Semb, H.: p120ctn-mediated organ patterning precedes and determines pancreatic progenitor fate. Developmental cell **49**(1), 31–47 (2019)

[30] Oord, A.v.d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K.: Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499 (2016)

[31] Pan, F.C., Wright, C.: Pancreas organogenesis: from bud to plexus to gland. Developmental Dynamics **240**(3), 530–565 (2011)

[32] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)

[33] Settles, B.: Active learning literature survey (2009)

[34] Shannon, C.E.: A mathematical theory of communication. ACM SIGMOBILE mobile computing and communications review **5**(1), 3–55 (2001)

[35] Shapiro, A.J., Lakey, J.R., Ryan, E.A., Korbutt, G.S., Toth, E., Warnock, G.L., Kneteman, N.M., Rajotte, R.V.: Islet transplantation in seven patients with type 1 diabetes mellitus using a glucocorticoid-free immunosuppressive regimen. New England Journal of Medicine **343**(4), 230–238 (2000)

[36] Shih, H.P., Wang, A., Sander, M.: Pancreas organogenesis: from lineage determination to morphogenesis. Annual review of cell and developmental biology **29**, 81–105 (2013)

[37] Skilling, J.: Nested sampling for general bayesian computation. Bayesian analysis **1**(4), 833–859 (2006)

[38] Skilling, J.: Bayesian computation in big spaces-nested sampling and galilean monte carlo. In: AIP Conference Proceedings 31st. vol. 1443, pp. 145–156. American Institute of Physics (2012)

[39] Skilling, J.: Galilean and hamiltonian monte carlo. In: Multidisciplinary Digital Publishing Institute Proceedings. vol. 33, p. 19 (2019)

[40] Speagle, J.S.: dynesty: a dynamic nested sampling package for estimating bayesian posteriors and evidences. Monthly Notices of the Royal Astronomical Society **493**(3), 3132–3158 (2020)

[41] Strzałko, J., Grabski, J., Stefański, A., Perlikowski, P., Kapitaniak, T.: Dynamics of coin tossing is predictable. Physics reports **469**(2), 59–92 (2008)

[42] Su, J., Vargas, D.V., Sakurai, K.: One pixel attack for fooling deep neural networks. IEEE Transactions on Evolutionary Computation **23**(5), 828–841 (2019)

[43] Sun, C., Shrivastava, A., Singh, S., Gupta, A.: Revisiting unreasonable effectiveness of data in deep learning era. In: Proceedings of the IEEE international conference on computer vision. pp. 843–852 (2017)

[44] Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning. pp. 6105–6114. PMLR (2019)

[45] Tanner, M.A.: Tools for statistical inference. Springer (2012)

[46] Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. Journal of machine learning research **2**(Nov), 45–66 (2001)

[47] Tschandl, P., Rosendahl, C., Kittler, H.: The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Scientific data **5**(1), 1–9 (2018)

[48] Villasenor, A., Chong, D.C., Henkemeyer, M., Cleaver, O.: Epithelial dynamics of pancreatic branching morphogenesis. Development **137**(24), 4295–4305 (2010)

[49] Yang, Y., Loog, M.: A benchmark and comparison of active learning for logistic regression. Pattern Recognition **83**, 401–415 (2018)

List of Publications

Appendix A Kasra Arnavaz, Aasa Feragen, Oswin Krause, and Marco Loog. "Bayesian Active Learning for Maximal Information Gain on Model Parameters" International Conference on Pattern Recognition (ICPR) 2019.

Appendix B Steffen Czolbe*, Kasra Arnavaz*, Oswin Krause, and Aasa Feragen "Is segmentation uncertainty useful" Information Processing in Medical Imaging (IPMI) 2021. (*Authors contributed equally.)

Appendix C Kasra Arnavaz, Pia Nyeng, Jelena M. Krivokapic, Oswin Krause, Aasa Feragen. "U-net segmentation of tubular structures from live imaging confocal microscopy: Successes and challenges" Virtual Early Career European Microscopy Congress (EMC) 2020.

Appendix D Kasra Arnavaz, Oswin Krause, Kilian Zepf, Jakob Andreas Bærentzen, Jelena M. Krivokapic, Silja Heilmann, Pia Nyeng, and Aasa Feragen. "Quantifying Topology In Pancreatic Tubular Networks From Live Imaging 3D Microscopy" under review at Machine Learning for Biomedical Imaging (MELBA) 2021.

Appendix A

Bayesian Active Learning for Maximal Information Gain on Model Parameters

The following chapter presents (up to formatting) the article

> Kasra Arnavaz, Aasa Feragen, Oswin Krause, and Marco Loog "Bayesian Active Learning for Maximal Information Gain on Model Parameters" International Conference on Pattern Recognition (ICPR) 2019.

This work investigates whether it is possible to find an active data gathering strategy which is guaranteed to outperform random sampling, or find a set of conditions under which this holds true. This work has been motivated by the segmentation problem that is the main topic of this thesis. While there have been sensible heuristics in the literature for active learning, we have no way of foreseeing whether a particular sampling strategy would be helpful for a given model and a data set. This hinders the usage of active learning for practical applications such as this one in this project.

We take a Bayesian inference approach to fitting model parameters, and use change in entropy to select data points that are expected to be most informative for the posterior over model parameters. We derive our sampling strategy for a logistic regression model, even though this line of thought could in principle be applied to any model. Since the posterior distribution for a logistic regression model is unimodal, it can be well approximated with a Gaussian distribution. This makes it possible to compute the entropy which we need to select samples by.

A drawback of the Bayesian approach is the assumption that the hypothesis space is correct, while Bayesian hypothesis testing is only through comparing different hypotheses [25]. Our derivation assumes that the model is correct, and then looks for most informative data to learn its plausible parameters, but if there were no set of parameters that could have given rise to the data, then searching for data points to fit its parameters seems like a lost cause. Our experiments fully reflect this issue.

Bayesian Active Learning for Maximal Information Gain on Model Parameters

Kasra Arnavaz[1,2], Aasa Feragen[3,1], Oswin Krause[1], and Marco Loog[4]

[1] Department of Computer Science, University of Copenhagen
{kasra, oswin.krause}@di.ku.dk
[2] DanStem, University of Copenhagen
afhar@dtu.dk
[3] DTU Compute, Technical University of Denmark
[4] Pattern Recognition Laboratory, Delft University of Technology
m.loog@tudelft.nl

Abstract. The fact that machine learning models, despite their advancements, are still trained on randomly gathered data is proof that a lasting solution to the problem of optimal data gathering has not yet been found. In this paper, we investigate whether a Bayesian approach to the classification problem can provide assumptions under which one is guaranteed to perform at least as good as random sampling. For a logistic regression model, we show that maximal expected information gain on model parameters is a promising criterion for selecting samples, assuming that our classification model is well-matched to the data. Our derived criterion is closely related to the maximum model change. We experiment with data sets which satisfy this assumption to varying degrees to see how sensitive our performance is to the violation of our assumption in practice.

1 Introduction

The default procedure to train a machine learning model is to learn from randomly gathered data. Active learning investigates whether we could reach at least the same performance as random sampling with fewer samples if we had the control over which samples to gather. While it might seem intuitive that the answer to the previous question should be positive, a consistent solution has been elusive so far [8, 25] . There have been several heuristics that propose sampling strategies based on common sense [20], and while they do outperform random sampling on occasion, it is not clear ahead of time if they will. What is missing is the set of conditions under which one is guaranteed to perform at least no worse than random sampling. This has been a hurdle which has prevented active learning strategies from replacing random sampling altogether.

In this paper, we employ a Bayesian framework for both model fitting as well as active learning to investigate the possibility of optimal data gathering—at least under certain assumptions. Parameters of our classification model are inferred through the approximation of the posterior distribution of parameters

by a Gaussian distribution. We, in turn, look for data which maximally reduces the expected entropy of this Gaussian posterior. Our derivation follows that of MacKay [12] for regression problems. We show that in the case of logistic regression, our sampling strategy has a nice, analytical form, which is closely related to maximum model change [2]. Moreover, in the limit of infinite data our sampling strategy behaves similarly to the "decision boundary sampling" [6]. We illustrate the behavior of our sampling strategy on a number of data sets, and in this context discuss how our derivations rely on our model being well-matched to the data. Having said that, we have no Bayesian way of quantifying the extent to which the model is well-matched to the data. Bayesian model testing is only able to compare different proposed models, and is in principle unable to detect if all those models are far from the truth. We conclude with a discussion of how this problem might be alleviated by extending the logistic regression model to a neural network.

The outline of this paper is as follows: Section 2 reviews how inference and active learning fit into the data modeling process, which we derive in detail in Section 3 and Section 4, respectively, following the analogous derivations of [9] and [12] for regression models. Particularly, in Subsection 3.1 we apply Laplace's method to Bayesian inference of model parameters and in Subsection 3.2 we address inference of a hyperparameter in an evidence maximization framework. In Section 4, we briefly review measures of information gain and introduce an active learning strategy to gain maximal information on model parameters. A performance criterion that incorporates parameter uncertainty is given in Section 5. In Section 6, we experimentally validate our active learning scheme on a range of data sets which satisfy our data set assumptions to varying degrees. Finally, Section 7 draws on our findings to discuss if any guarantees can be given about the optimality of our proposed method. We end by suggesting how the same line of thought could potentially lead to a better solution.[5]

2 Data Modeling

Before we delve into details, we find it of great value to remind the reader of the role of inference and data gathering in data modelling process [1]. We, as the data scientist, typically have a few candidate models (hypotheses), one of which we postulate underlies the data fairly well. These models might also include some parameters which we would like to set. Bayesian inference is the tool that enables us to reliably compare models and fit their parameters, taking into account the information produced by the data and our prior knowledge about which model or parameter is more likely. As a result, one could think of the following scenario for optimal data gathering [9]: Firstly, we may look for data that gives us maximal expected information on the plausibility of our candidate models. We may use this information to come up with new models we now find likely. We may continue this iteration until we have narrowed down our hypotheses to one. Once we have decided on one particular model, we would then aim to gather data which gives

[5] The code is available at `https://github.com/kasra-arnavaz/Bayesian-Active-Learning`.

us maximal expected information on the parameters of that particular model. A criterion that maximizes the discrimination of two models has been given in [12].

In this paper, we assume we have identified the so-called 'true model', and thus look for data that reduces the expected uncertainty of parameters the most. This assumption will incur consequences which we will discuss in Section 7.

3 Bayesian inference

Active learning goes hand in hand with uncertainty quantification. Bayesian inference provides a coherent basis for uncertainty quantification, where uncertainties are expressed by probability distributions. We refer the reader to [9] for an in-depth review of Bayesian inference for regression problems. Below, we will specifically work with the logistic regression model. This choice is made in part because it leads to analytical derivations, and in part because it is commonly used as the final output of more flexible deep learning classification networks.

3.1 Inference of model parameters w

In discriminative modeling, we are interested in finding a mapping from the input space to the target space which generalizes well to unseen data.

Suppose we have observed N input-target pairs as $\mathcal{D} = \{\mathbf{x}_n, t_n\}$, where $\mathbf{x}_n \in \mathbb{R}^k$, $t_n \in \{0, 1\}$, and $n = 1, ..., N$. A parametric model with parameters $\mathbf{w} \in \mathbb{R}^k$ is specified by its functional form $y(\mathbf{x}; \mathbf{w})$, the likelihood distribution $P(\{t_n\}|\{\mathbf{x}_n\}, \mathbf{w})$[6], and the prior distribution $P(\mathbf{w})$[7]. The functional form specifies a space of functions $\mathcal{F}$ through which one can wander by changing the parameters. The likelihood determines which functions in $\mathcal{F}$ fit the observed N samples better than the others. Prior establishes our prior belief about the plausibility of functions in $\mathcal{F}$. As our observed samples might also include outliers, prior usually takes a form which favors smooth functions to prevent the model from fitting to outliers and facilitate generalization. Our posterior belief of parameters would then be given by Bayes' theorem as

$$P(\mathbf{w}|\mathcal{D}) = \frac{P(\mathcal{D}|\mathbf{w})P(\mathbf{w})}{P(\mathcal{D})}. \tag{1}$$

One common prior which gives a varying degree of smoothness is a zero-mean Gaussian prior with variance $1/\alpha$ written as

$$P(\mathbf{w}|\alpha) = \frac{1}{Z_E} \exp\left(-\alpha E(\mathbf{w})\right), \tag{2}$$

[6] For notational convenience, from here onwards, we will write $P(\{t_n\}|\{x_n\}, \mathbf{w})$ as $P(\mathcal{D}|\mathbf{w})$, taking into account that in discriminative models we model targets given inputs.

[7] The prior could also depend on nuisance parameters, which we are momentarily assuming to be already integrated out.

where

$$E(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} \tag{3}$$

and

$$Z_E = \sqrt{(2\pi/\alpha)^k}. \tag{4}$$

For small α, the variance would be large, making the prior more like a uniform distribution and thus prior takes a neutral position in the posterior belief of the parameters in (1). In this case, parameters which fit the observed data better would be more plausible. Conversely, large α leads to a small variance, leaving the posterior to be dominated by the prior. In this case, over-smooth functions would be more plausible which might not fit the observed data well. In Subsection 3.2, we will apply Bayes' theorem to find which values of hyperparameter α are more likely.

We let the output of our model to estimate the probability of the positive class i.e. $P(t_n = 1|\mathbf{x}_n, \mathbf{w}) = y(\mathbf{x}_n; \mathbf{w})$. Consequently, we would get $P(t_n = 0|\mathbf{x}_n, \mathbf{w}) = 1 - y(\mathbf{x}_n; \mathbf{w})$. Therefore the probability of observed data would be given by

$$\begin{aligned} P(\mathcal{D}|\mathbf{w}) &= \prod_n y_n(\mathbf{w})^{t_n}(1 - y_n(\mathbf{w}))^{1-t_n} \\ &= \exp(-G(\mathbf{w})), \end{aligned} \tag{5}$$

where $y_n(\mathbf{w}) \equiv y(\mathbf{x}_n; \mathbf{w})$ and

$$G(\mathbf{w}) = -\sum_n t_n \log y_n(\mathbf{w}) + (1 - t_n)\log(1 - y_n(\mathbf{w})), \tag{6}$$

which is referred to as the binary-cross entropy loss.

By Bayes' theorem[8]

$$P(\mathbf{w}|\mathcal{D}, \alpha) = \frac{P(\mathcal{D}|\mathbf{w})P(\mathbf{w}|\alpha)}{P(\mathcal{D}|\alpha)}, \tag{7}$$

the posterior distribution over $\mathbf{w}$ can be written as

$$P(\mathbf{w}|\mathcal{D}, \alpha) = \frac{1}{Z_M}\exp(-M(\mathbf{w})), \tag{8}$$

where

$$M(\mathbf{w}) = G(\mathbf{w}) + \alpha E(\mathbf{w}), \tag{9}$$

and Z_M is the normalizing constant given by

$$Z_M = \int \exp(-M(\mathbf{w}))d^k\mathbf{w}. \tag{10}$$

[8] Data's dependence on α is only through $\mathbf{w}$ i.e. $P(\mathcal{D}|\mathbf{w}, \alpha) = P(\mathcal{D}|\mathbf{w})$.

This integral is intractable and we need the value of Z_M to infer α. To get around this problem, we can substitute $M(\mathbf{w})$ by its quadratic Taylor approximation around $\mathbf{w}_{\text{MAP}} = \text{argmin}M(\mathbf{w})$ as

$$M(\mathbf{w}) \simeq M(\mathbf{w}_{\text{MAP}}) + \frac{1}{2}(\Delta\mathbf{w}^T \mathbf{A} \Delta\mathbf{w}), \tag{11}$$

where

$$\Delta\mathbf{w} = \mathbf{w} - \mathbf{w}_{\text{MAP}}, \tag{12}$$

and

$$\mathbf{A} = \mathbf{H}_M(\mathbf{w}_{\text{MAP}}), \tag{13}$$

which is the Hessian matrix of $M(\mathbf{w})$ computed at $\mathbf{w}_{\text{MAP}}$. Consequently, the posterior distribution would be a Gaussian as

$$P(\mathbf{w}|\mathcal{D}, \alpha) = \frac{1}{Z_M} \exp(-\frac{1}{2}\Delta\mathbf{w}^T \mathbf{A} \Delta\mathbf{w}), \tag{14}$$

where

$$Z_M = \mathrm{e}^{-M(\mathbf{w}_{\text{MAP}})} \sqrt{(2\pi)^k / \det(\mathbf{A})}. \tag{15}$$

For a logistic regression model defined by

$$y_n(\mathbf{w}) := \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x}_n)}, \tag{16}$$

it can be verified that

$$\mathbf{A} = \sum_n y_n(\mathbf{w}_{\text{MAP}}) \left[1 - y_n(\mathbf{w}_{\text{MAP}})\right] \mathbf{x}_n \mathbf{x}_n^T + \alpha \mathbf{I}_k. \tag{17}$$

In summary, we approximated our posterior distribution by a Gaussian distribution with mean $\mathbf{w}_{\text{MAP}}$ and covariance matrix $\mathbf{A}^{-1}$.

3.2 Inference of hyperparameter α

The mean and the covariance matrix of the posterior distribution over $\mathbf{w}$ depend on α. To address this issue, we must take the expectation of $P(\mathbf{w}|\mathcal{D}, \alpha)$ when α's are drawn from their own posterior distribution $P(\alpha|\mathcal{D})$, or simply marginalize $P(\mathbf{w}, \alpha|\mathcal{D})$ over α i.e.

$$P(\mathbf{w}|\mathcal{D}) = \int P(\mathbf{w}|\mathcal{D}, \alpha) P(\alpha|\mathcal{D}) \, d\alpha. \tag{18}$$

The posterior over α is determined by

$$P(\alpha|\mathcal{D}) \propto P(\mathcal{D}|\alpha) P(\alpha). \tag{19}$$

Assuming a uniform prior over $\log \alpha$—since α appeared as an exponent in (2)—posterior belief over α is completely determined by its likelihood function $P(\mathcal{D}|\alpha)$

which is also known as the evidence for α. Evidence $P(\mathcal{D}|\alpha)$ appeared as the normalizing constant in (7) and is thus given by $P(\mathcal{D}|\alpha) = Z_M/Z_E$ which using (4) and (15) results in

$$\log P(\mathcal{D}|\alpha) = -M(\mathbf{w}_{\text{MAP}}) - \frac{1}{2}\log\det\mathbf{A} + \frac{k}{2}\log\alpha. \tag{20}$$

In the above equation, $M(\mathbf{w}_{\text{MAP}})$ and $\mathbf{A}$ depend on α according to (9) and (17). If we rewrite (17) as $\mathbf{A} = \mathbf{B} + \alpha\mathbf{I}_k$ and plug that together with (9) into (20), we get

$$\begin{aligned}\log P(\mathcal{D}|\alpha) = &- G(\mathbf{w}_{\text{MAP}}) - \alpha E(\mathbf{w}_{\text{MAP}}) \\ &- \frac{1}{2}\log\det(\mathbf{B} + \alpha\mathbf{I}_k) + \frac{k}{2}\log\alpha.\end{aligned} \tag{21}$$

We apply the same Gaussian approximation technique in the previous subsection to $P(\alpha|\mathcal{D})$, with the difference that we replace the Gaussian distribution by a delta function at its peak and keep track of the variance only as a measure of how dominant the peak is.

If $P(\alpha|\mathcal{D})$ has a dominant peak[9] at α_{MAP}, we can approximate (18) by

$$P(\mathbf{w}|\mathcal{D}) \simeq P(\mathbf{w}|\mathcal{D}, \alpha_{\text{MAP}}). \tag{22}$$

If we take the derivative of (21) w.r.t $\log\alpha$ and set it to zero, we find α_{MAP} will satisfy

$$2\alpha_{\text{MAP}}E(\mathbf{w}_{\text{MAP}}) + \alpha_{\text{MAP}}\text{Trace}((\mathbf{B} + \alpha_{\text{MAP}}\mathbf{I}_k)^{-1}) - k = 0. \tag{23}$$

We will solve this equation numerically to find α_{MAP}. Taking the second derivative of (21) w.r.t $(\log\alpha)^2$ computed at α_{MAP} shows how dominant the peak is and is given by [14]

$$\sigma_{\log\alpha|\mathcal{D}} \simeq \sqrt{\frac{2}{k - \alpha\text{Trace}(\mathbf{A}^{-1})}}. \tag{24}$$

With few labeled samples, $\alpha\text{Trace}(A^{-1})$ would be close to its maximum value k, which in turn leads to a large $\sigma_{\log\alpha|\mathcal{D}}$, so the peak would not be dominant and approximation (22) would not hold.

To conclude this section, our posterior distribution over model parameters, which also represents our uncertainty of parameters, is approximated to $P(\mathbf{w}|\mathcal{D}, \alpha_{\text{MAP}})$. Details on how we actually compute the posterior will be given in Section 6.

4 Bayesian Active Learning

Having gathered N input-target pairs, we would like to select the next input $\mathbf{x}_{N+1}$ such that we expect maximal information gain on model parameters once

[9] Empirically, if the model is well-matched to the data, this distribution is unimodal [9].

we receive target t_{N+1}. We will introduce two measures of information gain, both of which depend on entropy.

Entropy was originally introduced by Shannon [21] for discrete random variables. It measures how uncertain a discrete random variable is. For example, if we have a bent coin with bias p, the entropy is zero when $p = 0$ or $p = 1$ and is maximum when $p = 0.5$. For continuous random variables, entropy on its own is incompetent of conveying anything meaningful [16], mainly due to the fact that it is scale variant. However, change in entropy can be one measure of information gain (or loss).

Let us denote the probability distributions of parameters before and after we receive the target t_{N+1} by $P_N(\mathbf{w})$ and $P_{N+1}(\mathbf{w})$, respectively. Then information gain would mean a positive $\Delta S = S_N - S_{N+1}$, where

$$S_N = \int P_N(\mathbf{w}) \log \frac{1}{P_N(\mathbf{w})} \, d^k\mathbf{w} \tag{25}$$

is the entropy of the probability distribution of parameters before receiving t_{N+1}. Since the value of t_{N+1} is unknown to us when selecting $\mathbf{x}_{N+1}$, we will be working with the expectation over $P(t|\mathbf{x}, \mathcal{D})$ of our selected information gain measure. Another measure for information gain is the cross entropy between $P_N(\mathbf{w})$ and $P_{N+1}(\mathbf{w})$ defined as

$$C = \int P_{N+1}(\mathbf{w}) \log \frac{P_N(\mathbf{w})}{P_{N+1}(\mathbf{w})} \, d^k\mathbf{w}. \tag{26}$$

It is shown in [12] that change in entropy and cross entropy are equivalent in expectation i.e. $\mathbb{E}[\Delta S] = \mathbb{E}[C]$. Merely out of convenience, we will be working with change in entropy moving forward.

If we denote our entire training set by Q, from which N samples have been labeled, then our next query to label would be

$$\mathbf{x}_{N+1} = \underset{\mathbf{x} \in Q}{\operatorname{argmax}} \left(\mathbb{E}_{P(t|\mathbf{x},\mathcal{D})} \left[S_N - S_{N+1} \right] \right). \tag{27}$$

We approximated our posterior distribution over $\mathbf{w}$ by a Gaussian in (14). It can be shown that the entropy of a k-dimensional Gaussian distribution with covariance matrix $\mathbf{A}^{-1}$ is [12]

$$S = \frac{k}{2}(1 + \log 2\pi) + \frac{1}{2} \log(\det \mathbf{A}^{-1}). \tag{28}$$

Therefore change in entropy would equal to

$$\Delta S = \frac{1}{2} \log \frac{\det \mathbf{A}_{N+1}}{\det \mathbf{A}_N}. \tag{29}$$

Due to (17), the relationship between $\mathbf{A}_{N+1}$ and $\mathbf{A}_N$ is

$$\mathbf{A}_{N+1} = \mathbf{A}_N + y_{N+1}(\mathbf{w}_{\text{MAP}}) \left[1 - y_{N+1}(\mathbf{w}_{\text{MAP}}) \right] \mathbf{x}_{N+1} \mathbf{x}_{N+1}^T. \tag{30}$$

For a scalar β and a vector $\mathbf{x}$, determinant has the property $\det\left[\mathbf{A}+\beta\mathbf{x}\mathbf{x}^T\right] = (\det\mathbf{A})(1+\beta\mathbf{x}^T\mathbf{A}^{-1}\mathbf{x})$. Applying this property to (30), we can rewrite (29) as

$$\Delta S = \frac{1}{2}\log(1+m), \tag{31}$$

where

$$m = y_{N+1}(\mathbf{w}_{\mathrm{MAP}})\left[1-y_{N+1}(\mathbf{w}_{\mathrm{MAP}})\right]\mathbf{x}_{N+1}^T\mathbf{A}_N^{-1}\mathbf{x}_{N+1}. \tag{32}$$

Note that this expression is independent of t_{N+1}, and as a result $\mathbb{E}(\Delta S) = \Delta S$. This is a mere consequent of choosing a logistic regression model, and might not hold for other models.

We refer to the criterion defined by (31) and (32), as maximal expected information gain or just information gain for short. This criterion has a nice property that $y_{N+1}(\mathbf{w}_{\mathrm{MAP}})\,(1-y_{N+1}(\mathbf{w}_{\mathrm{MAP}}))$ favors the points close to the decision boundary while $\mathbf{x}_{N+1}^T\mathbf{A}_N^{-1}\mathbf{x}_{N+1}$ favors the points on the far end of data space. In particular, for small N, where we have a high uncertainty over parameters, $\mathbf{x}_{N+1}^T\mathbf{A}_N^{-1}\mathbf{x}_{N+1}$ would be the dominating term, so the criterion would select points which have a larger norm and lie close to the decision boundary. As we gather more data, eigenvalues of the covariance $\mathbf{A}_N^{-1}$ would get smaller, and $y_{N+1}(\mathbf{w}_{\mathrm{MAP}})\,(1-y_{N+1}(\mathbf{w}_{\mathrm{MAP}}))$ would be the dominating term in the criterion. At this point, maximal expected information gain behaves close to decision boundary sampling in [6]. Figure 1 illustrates this intuition of the two sampling strategies for $N = 2, 7, 12, 17$ and 22.

Lastly, information gain criterion is closely related to maximum model change criterion in [2] for a logistic regression model. Inspired by the gradient descent update rule, which updates parameters in the opposite direction of the gradient of the loss function, the authors in [2] propose a sampling strategy which maximizes the expected gradient length of the loss function.

5 Bayesian prediction

By the application of probability rules, Bayesian prediction takes the uncertainty of parameters into account when predicting new targets. For an input $\mathbf{x}$, our prediction for its corresponding target t to belong to the positive class is the expected value of our model output when the parameters are drawn from the posterior distribution i.e.

$$P(t=1|\mathbf{x},\mathcal{D}) = \int P(t=1|\mathbf{x},\mathbf{w})P(\mathbf{w}|\mathcal{D})\;d^k\mathbf{w}. \tag{33}$$

The term $P(t=1|\mathbf{x},\mathcal{D})$ is written in shorthand by $y(\mathbf{x})$ and is referred to as marginalized output.

Under our current assumptions $P(\mathbf{w}|\mathcal{D})$ is a Gaussian distribution according to (14), and $P(t=1|\mathbf{x},\mathbf{w})$ is a sigmoidal function according to (16), which render

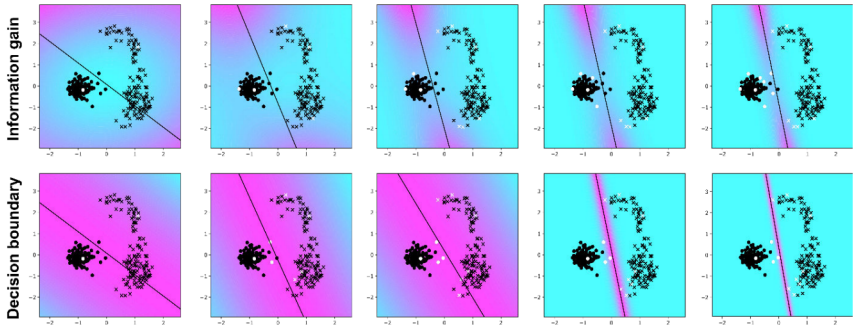


Fig. 1: Here, we show the sampling procedure for maximal expected information gain versus decision boundary sampling for a 2D version of the Digits data set obtained by projecting it onto its first two principal components. We show the points drawn at iteration 0, 5, 10, 15, and 20, respectively. The black dots are unlabeled samples and the white ones are labeled samples. The color indicates the value (pink=high, blue=low) of the property being maximized, i.e. the information gain (top) and the distance to decision boundary (bottom). Note how samples are drawn near pink areas.

the above integral intractable. Maximum a Posteriori (MAP) method estimates this integral by

$$P(t=1|\mathbf{x},\mathcal{D}) \simeq P(t=1|\mathbf{x},\mathbf{w}_{\text{MAP}}), \tag{34}$$

which is equivalent of replacing the posterior $P(\mathbf{w}|\mathcal{D})$ by a delta function at its peak $\mathbf{w}_{\text{MAP}}$ (similar to what we did for the inference of α). A better approximation has been in suggested [11] resulting in a predictive distribution as [10]

$$P(t=1|\mathbf{x},\mathcal{D}) = \frac{1}{1+\exp(-\mathbf{w}_{\text{MAP}}^T\mathbf{x}/\sqrt{1+\frac{\pi}{8}\mathbf{x}^T\mathbf{A}^{-1}\mathbf{x}})}. \tag{35}$$

This equation comes with a nice interpretation: When the uncertainty of parameters are low i.e. the eigenvalues of the covariance $\mathbf{A}^{-1}$ are small, we get $\mathbf{x}^T\mathbf{A}^{-1}\mathbf{x} \to 0$, which implies $P(t=1|\mathbf{x},\mathcal{D}) \to P(t=1|\mathbf{x},\mathbf{w}_{\text{MAP}})$, meaning MAP estimation is valid. On the other hand, when the eigenvalues (uncertainties) are large, giving $\mathbf{x}^T\mathbf{A}^{-1}\mathbf{x} \to \infty$, we see that $P(t=1|\mathbf{x},\mathcal{D}) \to 0.5$, meaning that we have low confidence in our predictions due to high uncertainty in parameters. MAP estimation is no longer valid in this case. In conclusion, the term $\mathbf{x}^T\mathbf{A}^{-1}\mathbf{x}$ prevents overconfident predictions when the model has been exposed to little data and is uncertain about the parameters as a result.

Having settled our prediction, we would seek to define measures for out-of-sample performance on a given test set $\{\mathbf{x}_m, t_m | m = 1, \cdots, M\}$. Accuracy defined by (37) is ignorant of prediction uncertainty and is not preferred as a

[10] This approximation becomes inaccurate when $\mathbf{w}_{\text{MAP}}^T\mathbf{x} \gg \mathbf{x}^T\mathbf{A}^{-1}\mathbf{x} \gg 1$.

result.

$$\text{accuracy} = \frac{1}{M}\sum_m \mathbb{1}(t_m = \lfloor y_m \rceil), \tag{36}$$

where $\mathbb{1}(\cdot)$ outputs one if its argument is True and zero if False, $\lfloor\cdot\rceil$ rounds its argument to closest integer, and $y_m \equiv y(\mathbf{x}_m)$.

An alternative would be the binary cross entropy loss between the targets t_m and marginalized outputs y_m, which we call marginalized loss, and is given by

$$\text{marginalized loss} = -\frac{1}{M}\sum_m t_m \log y_m + (1 - t_m)\log(1 - y_m). \tag{37}$$

Due to the intuition given for marginalized output, a model which makes a mistake in the classification of a test point but is unsure about its prediction incurs a lower marginalized loss, than the one which makes a mistake and is confident about its prediction.

With marginalized loss as our preferred measure of performance on unseen data, we head to experiment.

6 Experiments

We compare our maximal information gain strategy with two other sampling strategies using logistic regression; namely random sampling and decision boundary sampling [6], where the latter picks the closest possible sample to the decision boundary under the assumption that these are maximally uncertain. We perform experiments on the following 10 classification data sets: AB and ABA are synthetic data sets sampled from 2 and 3 2-dimensional Gaussian distributions, respectively, to produce data sets which are linearly separable (AB) and not linearly separable (ABA). The Breast Cancer Wisconsin (Diagnostic) [15], Optical Recognition of Handwritten Digits [24], Statlog (Heart), Haberman's Survival [5], Parkinson's [7], and Ionosphere [23] data sets come from the UCI repository [4]; here, for Digits only the first two classes were used. DD [3,22] and AIDS [19,26] are benchmark graph learning data sets [17], where each graph was given a vector representation via its node degree histogram. The details of the 10 data sets are shown in Table 1.

When training logistic regression, we pick the initial value of $\log\alpha$ according to its prior, $\mathcal{U}(10^{-3}, 10)$ in our case, each time a new training point is drawn. With this value of α, we minimize $M(\mathbf{w})$, and then update α according to (23). We plug in this new α into the equation for $M(\mathbf{w})$ and repeat this process until convergence, which is guaranteed if optimizations of $M(\mathbf{w})$ at each stage are not far off [14]. The final values for $\mathbf{w}_{\text{MAP}}$ and $\mathbf{A}$ will be used for the prediction.

When using the maximal expected information gain sampling, we start the experiments by randomly revealing one labeled data point from each class, and add in one sample at a time that maximizes (31). We repeat this setup 20 times changing the two revealed data points each time. A similar procedure is applied to the decision boundary sampling and random sampling strategies; these are

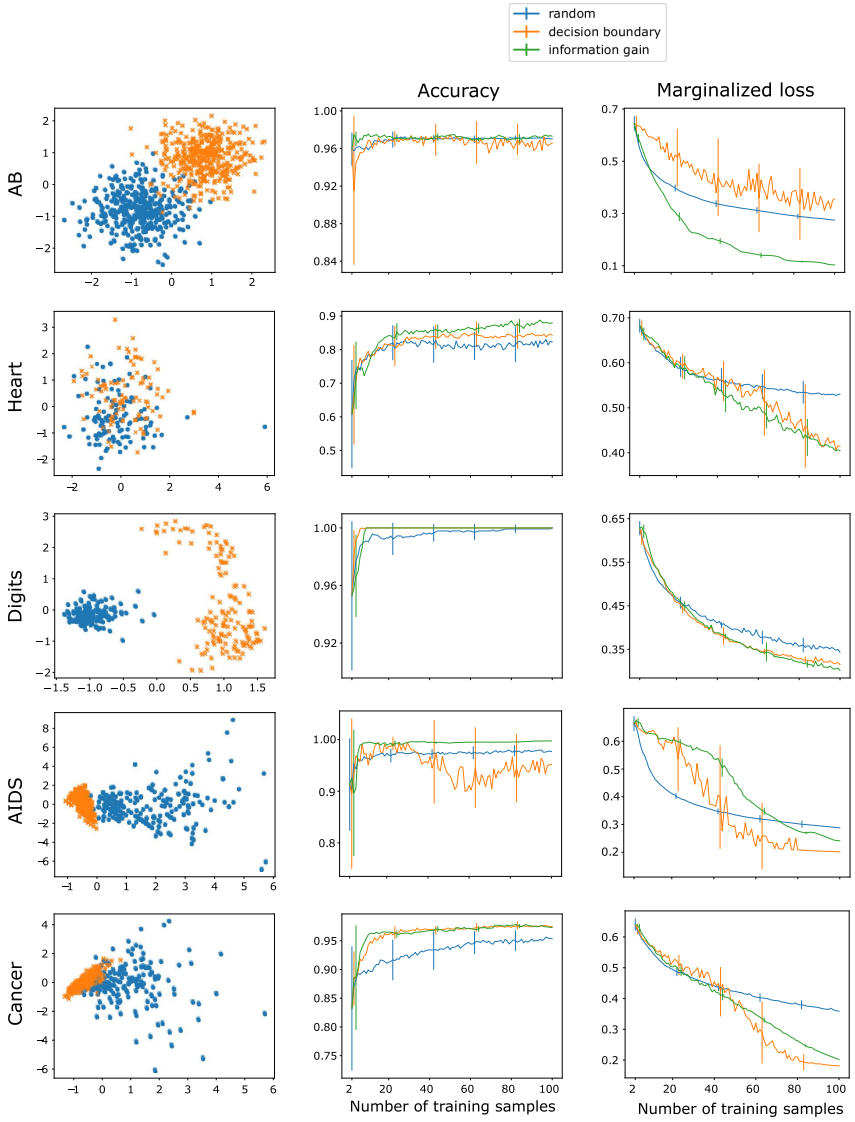


Fig. 2: For 5 roughly linearly separable data sets, we see (left) a visualization of the data set via projection onto the first two principal components, as well as the (middle) accuracy and (right) marginalized loss, both as a function of number of training samples seen. The plots contain the mean and standard deviations of 20 repeated runs of the sampling strategies. As the logistic regression model is well-matched to these data sets, the information gain criterion for gathering samples outperforms random sampling.

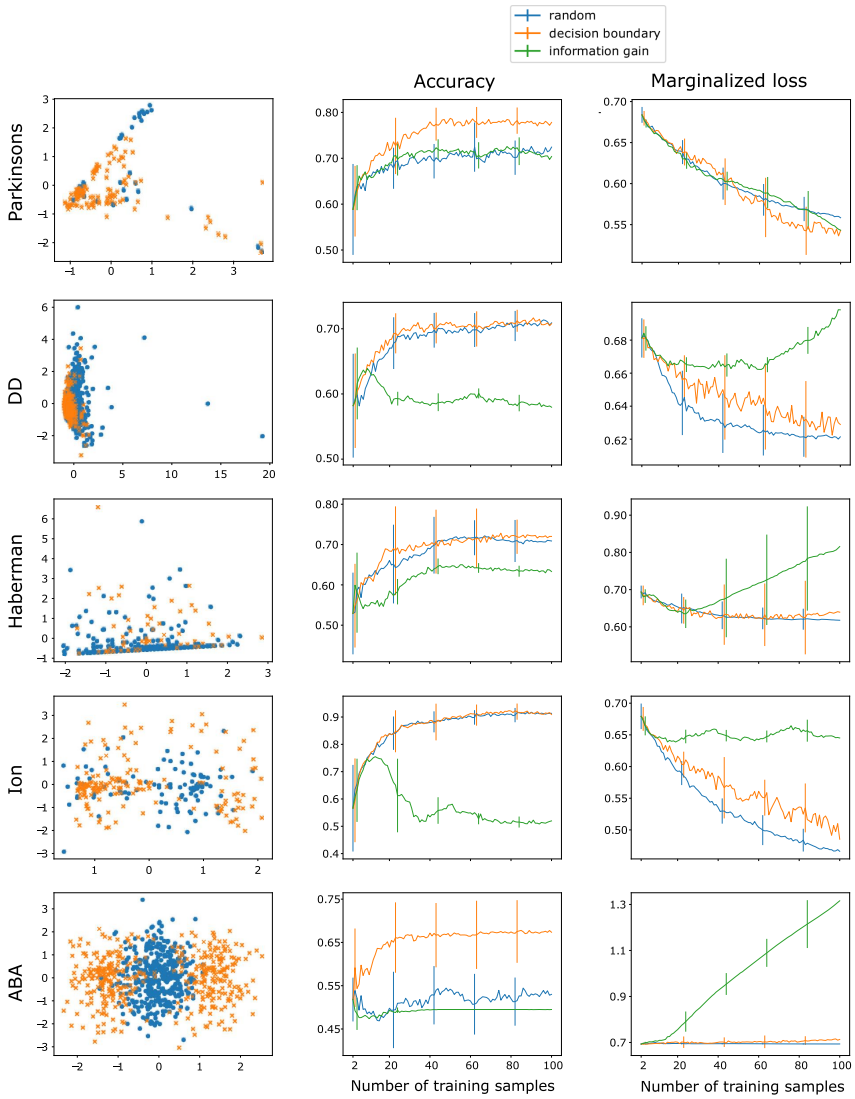


Fig. 3: For 5 not linearly separable data sets, we see (left) a visualization of the data set via projection onto the first two principal components, as well as the (middle) accuracy and (right) marginalized loss, both as a function of number of training samples seen. The plots contain the mean and standard deviations of 20 repeated runs of the sampling strategies. Since the assumption that our hypothesis space is correct no longer holds, the information gain criterion performs poorly.

Table 1: Data set details

Name	AB	Heart	Digits	AIDS	Cancer
# Samples	1000	270	360	2000	569
# Features	2	13	64	10	30

Name	Parkinson's	DD	Haberman	Ion	ABA
# Samples	195	1178	306	351	1000
# Features	22	19	3	34	2

initialized with the same two data points per run as the maximal information gain sampling. Samples are selected with replacement among all strategies i.e. relabeling of the same input is possible. Otherwise, all sampling strategies would reach the same performance when all samples in Q are exhausted [8].

Figure 2 and Figure 3 show the mean and standard deviation of the accuracy and marginalized loss over the different data sets for the three sampling strategies. The figures also show (left column) the projection of the data onto the first two principal components of each data set, to give insight into data set properties.

Data sets in Figure 2 compose of two clusters for each class which are roughly separable by a line. Even Heart data set looks like two Gaussians whose means are close to each other. Looking at their corresponding accuracies and marginalized loss, we see that at least in the limit of high amount of data, information gain outperforms random sampling consistently. When the number of training samples is small, in the cases where we obtain inferior performance compared to random sampling, we speculate the reason to be a large $\sigma_{\log \alpha|\mathcal{D}}$ given in (24); not to mention that approximation (35) breaks when $\mathbf{w}_{\text{MAP}}^T\mathbf{x} \gg \mathbf{x}^T\mathbf{A}^{-1}\mathbf{x} \gg 1$. It is visually visible that for data sets in Figure 3, however, the classes are less linearly separable to the point that ABA clearly is not. In such scenarios, information gain sampling does not perform satisfactorily in neither accuracy nor marginalized loss. The reason is that the logistic regression model is not well-matched to these data sets. Information gain criterion works by taking uncertainty of parameters into account, but the input-target relationship in these data sets cannot be modeled by a logistic regression, so the covariance matrix A^{-1} we compute in (17) is far from anything meaningful. This is verified by the increasing marginalized loss in DD, Haberman, and ABA. We will resume this discussion in the next section.

7 Discussion and Conclusion

We have derived an active learning sampling scheme for classification based on maximizing information gain on model parameters. Additionally, we have shown that in the case of logistic regression, this scheme takes a nice, interpretable

form and, in particular, is closely related to the more ad hoc maximum model change [2]. Via our experiments we also see how the performance of the method depends whether our choice of classifier is well matched to the data set, an assumption implicitly underlying our analysis.

In particular, in Section 2, we assumed that we have found the so-called true model, and as a result the purpose of data gathering is to infer the plausibility of model parameters with smallest possible amount of labeled data. Subsequently, we derived a criterion under the Bayesian framework assuming that the model is well-matched to the data. In Bayesian language, our assumption is that the hypothesis space is correct. Our experiments confirmed that indeed, when the hypothesis space is correct, the information gain criterion could be a promising sampling strategy—perhaps with more accurate approximations using Monte Carlo methods [18]. That is reassuring except the fact that we have no Bayesian way of verifying that our hypothesis space actually is correct [10].

Hypothesis testing in the Bayesian framework is performed through model-comparison. Let's say we have several hypotheses $\mathcal{H}_i$ whose validity we want to investigate, and we have gathered data $\mathcal{D}$. Thanks to Bayes' theorem

$$P(\mathcal{H}_i|\mathcal{D}) \propto P(\mathcal{D}|\mathcal{H}_i)P(\mathcal{H}_i), \tag{38}$$

we can sort alternative hypotheses in order of their plausibility. They could all be completely far from the truth and we would get a ranking regardless. Therefore, Bayesian model comparison is also only viable if we are within the correct hypothesis space, where Bayes wouldn't prefer other hypotheses to the truth [9].

It seems inevitable that one must err on the side of a larger hypothesis space. Logistic regression is essentially a classification neural network with no hidden layer, which has the nice property that its posterior over parameters with a Gaussian prior is unimodal. For a neural network with hidden layers, the posterior could be multimodal, and even if we can find the global maximum of the posterior, fitting a Gaussian around that point is not an acceptable substitute for the entire posterior distribution. However, a solution has been given in [13] which is to fit a Gaussian distribution around each local maximum of the posterior and treat each maximum separately. One can then use Bayes' theorem to compare them to one another. Although this might not be a permanent solution, since neural networks of a fixed width and depth are not still universal approximators, they nevertheless specify a larger hypothesis space than logistic regression. Extending our analysis to more flexible, modern classifiers remains an important avenue for future research.

Acknowledgment

This work was supported by the Novo Nordisk Foundation grant NNF17OC0028360.

References

1. Box, G.E., Tiao, G.C.: Bayesian inference in statistical analysis, vol. 40. John Wiley & Sons (2011)
2. Cai, W., Zhang, Y., Zhang, Y., Zhou, S., Wang, W., Chen, Z., Ding, C.: Active learning for classification with maximum model change. ACM Transactions on Information Systems (TOIS) **36**(2), 1–28 (2017)
3. Dobson, P.D., Doig, A.J.: Distinguishing enzyme structures from non-enzymes without alignments. Journal of molecular biology **330**(4), 771–783 (2003)
4. Dua, D., Graff, C.: UCI machine learning repository (2017), `http://archive.ics.uci.edu/ml`
5. Haberman, S.J.: Generalized residuals for log-linear models. In: Proceedings of the 9th international biometrics conference. pp. 104–122 (1976)
6. Lewis, D.D., Gale, W.A.: A sequential algorithm for training text classifiers. In: SIGIR'94. pp. 3–12. Springer (1994)
7. Little, M.A., McSharry, P.E., Roberts, S.J., Costello, D.A., Moroz, I.M.: Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. Biomedical engineering online **6**(1), 23 (2007)
8. Loog, M., Yang, Y.: An empirical investigation into the inconsistency of sequential active learning. In: 2016 23rd international conference on pattern recognition (ICPR). pp. 210–215. IEEE (2016)
9. MacKay, D.J.: Bayesian interpolation. Neural computation **4**(3), 415–447 (1992)
10. MacKay, D.J.: Bayesian methods for adaptive models. Ph.D. thesis, California Institute of Technology (1992)
11. MacKay, D.J.: The evidence framework applied to classification networks. Neural computation **4**(5), 720–736 (1992)
12. MacKay, D.J.: Information-based objective functions for active data selection. Neural computation **4**(4), 590–604 (1992)
13. MacKay, D.J.: A practical bayesian framework for backpropagation networks. Neural computation **4**(3), 448–472 (1992)
14. MacKay, D.J.: Comparison of approximate methods for handling hyperparameters. Neural computation **11**(5), 1035–1068 (1999)
15. Mangasarian, O.L., Wolberg, W.H.: Cancer diagnosis via linear programming. Tech. rep., University of Wisconsin-Madison Department of Computer Sciences (1990)
16. Marsh, C.: Introduction to continuous entropy. Department of Computer Science, Princeton University (2013)
17. Morris, C., Kriege, N.M., Bause, F., Kersting, K., Mutzel, P., Neumann, M.: Tu-dataset: A collection of benchmark datasets for learning with graphs. In: ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020). `www.graphlearning.io`
18. Neal, R.M.: Bayesian learning for neural networks, vol. 118. Springer Science & Business Media (2012)
19. Riesen, K., Bunke, H.: Iam graph database repository for graph based pattern recognition and machine learning. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) Structural, Syntactic, and Statistical Pattern Recognition. pp. 287–297 (2008)
20. Settles, B.: Active learning literature survey. Tech. rep., University of Wisconsin-Madison Department of Computer Sciences (2009)

21. Shannon, C.E.: A mathematical theory of communication. ACM SIGMOBILE mobile computing and communications review **5**(1), 3–55 (2001)
22. Shervashidze, N., Schweitzer, P., Van Leeuwen, E.J., Mehlhorn, K., Borgwardt, K.M.: Weisfeiler-lehman graph kernels. Journal of Machine Learning Research **12**(9) (2011)
23. Sigillito, V.G., Wing, S.P., Hutton, L.V., Baker, K.B.: Classification of radar returns from the ionosphere using neural networks. Johns Hopkins APL Technical Digest **10**(3), 262–266 (1989)
24. Xu, L., Krzyzak, A., Suen, C.Y.: Methods of combining multiple classifiers and their applications to handwriting recognition. IEEE transactions on systems, man, and cybernetics **22**(3), 418–435 (1992)
25. Yang, Y., Loog, M.: A benchmark and comparison of active learning for logistic regression. Pattern Recognition **83**, 401–415 (2018)
26. Zaharevitz, D.: Aids antiviral screen data (2015)

Appendix B

Is segmentation uncertainty useful?

The following chapter presents the article

> Steffen Czolbe*, Kasra Arnavaz*, Oswin Krause, and Aasa Feragen. "Is segmentation uncertainty useful?" Information Processing in Medical Imaging (IPMI) 2021. (*Authors contributed equally.)

This work studies established probabilistic segmentation models for two main purposes: 1. Do uncertainty estimates of these models reflect segmentation error? 2. Are the uncertainties provided by these models useful for active learning? Both of these questions are of major importance for many biomedical imaging applications including the topic of this thesis.

In particular, endowing predicted segmentations with a well-adjusted measure of reliability could help the biologists in determining which predictions are trustworthy to test their biological hypothesis on. Furthermore, annotating 3D images of such low signal-to-noise ratio is time-consuming. As a result, it would be cost effective if uncertainties provided by the probabilistic segmentation models could be used to carefully select a subset of samples to be annotated.

We compare four probabilistic segmentation models on two data sets: ISIC18 [22, 47] with a single annotator, and LIDC [2, 8] with four annotators. We find that on the ISIC18 data set all four models have a low uncertainty when making a true classification and a high uncertainty when making misclassification. For LIDC, the uncertainty is low when annotators agree and high when they disagree. As for the active learning, we select the samples which are closest to the decision boundary [21] and add them to our training set and compare this against a case where samples are added at random. On neither data set did we observe a consistent benefit for any of the four models studied.

We could not use the criterion suggested in Appendix A for active learning, because the posterior of such deep neural networks is a complex distribution, making it intractable to compute its entropy. Moreover, the posterior is most likely multimodal and approximating it with a single Gaussian is not possible (unless it is sharply peaked which normally does not occur under limited data regime of active learning).

Is segmentation uncertainty useful?

Steffen Czolbe[†1], Kasra Arnavaz[†1], Oswin Krause[1], and Aasa Feragen[2]

[1] University of Copenhagen, Department of Computer Science, Denmark,
per.sc,kasra,oswin.krause@di.ku.dk
[2] Technical University of Denmark, DTU Compute, Denmark,
afhar@dtu.dk
† Authors contributed equally

Abstract. Probabilistic image segmentation encodes varying prediction confidence and inherent ambiguity in the segmentation problem. While different probabilistic segmentation models are designed to capture different aspects of segmentation uncertainty and ambiguity, these modelling differences are rarely discussed in the context of applications of uncertainty. We consider two common use cases of segmentation uncertainty, namely assessment of segmentation quality and active learning. We consider four established strategies for probabilistic segmentation, discuss their modelling capabilities, and investigate their performance in these two tasks. We find that for all models and both tasks, returned uncertainty correlates positively with segmentation error, but does not prove to be useful for active learning.

Keywords: Image segmentation · Uncertainty quantification · Active learning.

1 Introduction

Image segmentation – the task of delineating objects in images – is one of the most crucial tasks in image analysis. As image acquisition methods can introduce noise, and experts disagree on ground truth segmentations in ambiguous cases, predicting a single segmentation mask can give a false impression of certainty. Uncertainty estimates inferred from the segmentation model can give some insight into the confidence of any particular segmentation mask, and highlight areas of likely segmentation error to the practitioner. It adds transparency to the segmentation algorithm and communicates this uncertainty to the user. This is particularly important in medical imaging, where segmentation is often used to understand and treat disease. Consequently, quantification of segmentation uncertainty has become a popular topic in biomedical imaging [6, 11].

Training segmentation networks requires large amounts of annotated data, which are costly and cumbersome to attain. Active learning aims to save the annotator's time by employing an optimal data gathering strategy. Some active

Code available at github.com/SteffenCzolbe/probabilistic_segmentation

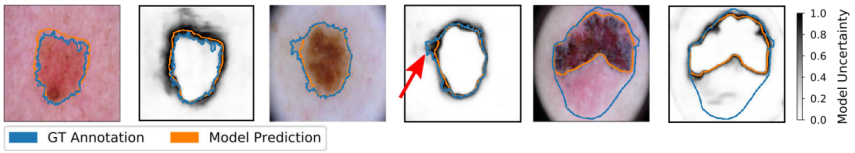


Fig. 1: Segmentation uncertainty is often interpreted as probable segmentation error, as seen near the lesion boundary in the first two examples. In the third example, however, model bias leads to a very certain, yet incorrect segmentation.

learning methods use uncertainty estimates to select the next sample to annotate [7,9,10]. While several potential such data gathering strategies exist [13,16], a consistent solution remains to be found [8].

While several methods have been proposed to quantify segmentation uncertainty [4,6,11], it is rarely discussed what this uncertainty represents, whether it matches the user's interpretation, and if it can be used to formulate a data-gathering strategy. We compare the performance of several well-known probabilistic segmentation algorithms, assessing the quality and use cases of their uncertainty estimates. We consider two segmentation scenarios: An unambiguous one, where annotators agree on one underlying true segmentation, and an ambiguous one, where a set of annotators provide potentially strongly different segmentation maps, introducing variability in the ground truth annotation.

We investigate the degree to which the inferred uncertainty correlates with segmentation error, as this is how reported segmentation uncertainty would typically be interpreted by practitioners. We find that uncertainty estimates of the models coincide with likely segmentation errors and strongly correlate with the uncertainty of a set of expert annotators. Surprisingly, the model architecture used does not have a strong influence on the quality of estimates, with even a deterministic U-Net [12] giving good pixel-level uncertainty estimates.

Second, we study the potential for uncertainty estimates to be used for selecting samples for annotation in active learning. Reducing the cost of data annotation is of utmost importance in biomedical imaging, where data availability is fast-growing, while annotation availability is not. We find that there are many pitfalls to an uncertainty-based data selection strategy. In our experiment with multiple annotators, the images with the highest model uncertainty were precisely those images where the annotators were also uncertain. Labeling these ambiguous images by a group of expert annotators yielded conflicting ground truth annotations, providing little certain evidence for the model to learn from.

2 Modelling segmentation uncertainty

Image segmentation seeks to estimate a well-defined binary[3] segmentation $g\colon \Omega \to \{0,1\}$ for a discrete image domain Ω. Typically, a predictive model $h(\mathbf{x},\mathbf{w})$ with

[3] For simplicity, we consider binary segmentation; the generalization to multi-class segmentation is straightforward.

parameters $\mathbf{w}$, such as a neural network, is fitted to binary annotation data $a\colon \Omega \to \{0,1\}$ by minimizing a loss $\mathcal{L}(a, h(\mathbf{x}, \mathbf{w}))$. Here, $\mathbf{x} \in \mathbb{R}^{\Omega}$ is the image, and $\mathbf{y} = h(\mathbf{x}, \mathbf{w})$ defines an image of pixel-wise segmentation probabilities, such as the un-thresholded softmax output of a segmentation network h.

Typically, the annotation is assumed to be error-free, that is $a = g$, and predictors are typically trained on a single annotation per image. We assume that the trained neural network $h(\mathbf{x}, \mathbf{w})$ satisfies

$$h(\mathbf{x}, \mathbf{w}) = g(\mathbf{x}) + b + err \ ,$$

where b and err denote bias and segmentation error. Segmentation uncertainty is often interpreted as correlating with this error, although this is primarily realistic for small bias. Such segmentation tasks are called *unambiguous*; we consider a running example of skin lesion segmentation from dermoscopic images [3, 15], where the lesion boundary is clearly visible in the image (Fig. 1).

Recent work has considered *ambiguous* segmentation tasks [6,11], where there is no accessible "ground truth" segmentation, either because the data is not sufficient to estimate the segmentation, or because there is subjective disagreement. Examples include lesions in medical imaging, where the boundary can be fuzzy due to gradual infiltration of tissue, or where experts disagree on whether a tissue region is abnormal or not.

In such tasks, we make no assumption on the underlying segmentation g or the errors err, but regard the observed annotations as samples from an unknown "ground truth" distribution $p(a|\mathbf{x})$ over annotations a conditioned on the image $\mathbf{x}$. The goal of segmentation is to estimate the distribution $p(a|\mathbf{x})$, or its proxy distribution $p(\mathbf{y}|\mathbf{x})$ over pixel-wise class probabilities $\mathbf{y}\colon \Omega \to [0,1]$, as accurately as possible for a given image $\mathbf{x}$. If successful, such a model can sample coherent, realistic segmentations from the distribution, and estimate their variance and significance. As a running example of an ambiguous segmentation task, we consider lung lesions [1, 2, 6]. For such tasks, predictors are typically trained on multiple annotators, who may disagree both on the segmentation boundary and on whether there is even an object to segment.

From the uncertainty modelling viewpoint, these two segmentation scenarios are rather different. Below, we discuss differences in uncertainty modelling for the two scenarios and four well-known uncertainty quantification methods.

3 Probabilistic Segmentation Networks

A probabilistic segmentation model seeks to model the distribution $p(\mathbf{y}|\mathbf{x})$ over segmentations given an input image $\mathbf{x}$. Here, our annotated dataset $(\mathbf{X}, \mathbf{A})$ consists of the set $\mathbf{X}$ of N images $\{\mathbf{x}_n \mid n = 1, ..., N\}$, and L annotations are available per image, so that $\mathbf{A} = \{a_n^{(l)} \sim p(\mathbf{y}|\mathbf{x}_n) \mid (n, l) = (1, 1), ..., (N, L)\}$.

Taking a Bayesian view, we seek the distribution

$$p(\mathbf{y}|\mathbf{x}, \mathbf{X}, \mathbf{A}) = \int p(\mathbf{y}|\mathbf{x}, \mathbf{w}) p(\mathbf{w}|\mathbf{X}, \mathbf{A}, h)\, d\mathbf{w} \ , \tag{1}$$

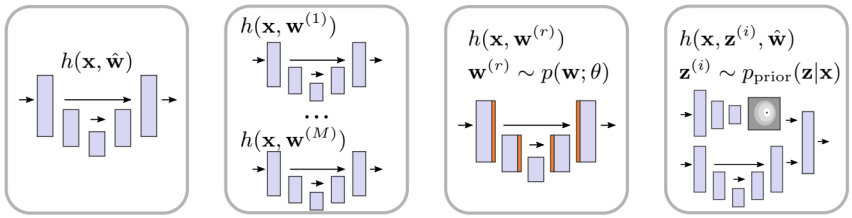


Fig. 2: Schematic overview (adapted from [6]) of the evaluated models. Blue: residual blocks . Orange: Dropout layers essential to the networks' functionality.

over segmentations $\mathbf{y}$ given image $\mathbf{x}$ and data $(\mathbf{X}, \mathbf{A})$, which can be obtained by marginalization with respect to the weights $\mathbf{w}$ of the model h.

In most deep learning applications, our prior belief over the model h, denoted $p(h)$, is modelled by a Dirac delta distribution indicating a single architecture with no uncertainty. In the context of uncertain segmentation models, however, we would like to model uncertainty in the parameters $\mathbf{w}$. Denoting our prior belief over the parameters $\mathbf{w}$ by $p(\mathbf{w}|h)$, Bayes' theorem gives

$$p(\mathbf{w}|\mathbf{X}, \mathbf{A}, h) = \frac{p(\mathbf{w}|h)p(\mathbf{A}|\mathbf{X}, \mathbf{w}, h)}{p(\mathbf{A}|\mathbf{X}, h)}, \tag{2}$$

where the likelihood update function is given by

$$p(\mathbf{A}|\mathbf{X}, \mathbf{w}, h) = \exp\left(\sum_{n=1}^{N}\sum_{l=1}^{L} \mathbf{A}^{(l)} \log\left(h(\mathbf{x}_n, \mathbf{w})\right) + (1 - \mathbf{A}^{(l)}) \log\left(1 - h(\mathbf{x}_n, \mathbf{w})\right)\right)$$

and normalizing constant

$$p(\mathbf{A}|\mathbf{X}, h) = \int p(\mathbf{w}|h)p(\mathbf{A}|\mathbf{X}, \mathbf{w}, h)\, d\mathbf{w} \ .$$

This integral is generally intractable, making it impossible to obtain the proper posterior (2). Below, we discuss how empirical approximations $\hat{p}(\mathbf{y}|\mathbf{x}, \mathbf{X}, \mathbf{A})$ to the distribution $p(\mathbf{y}|\mathbf{x}, \mathbf{X}, \mathbf{A})$ found in (1) are performed in four common segmentation models. Note that both p and $\hat{p}$ can be degenerate, depending on the number of annotations available and models used.

U-Net with softmax output. The well established U-Net [12] architecture with a softmax output layer yields class-likelihood estimates. As the model is deterministic, $p(h|\mathbf{X}, \mathbf{A})$ is degenerate. Parameters are selected by a maximum a posteriori (MAP) estimate i.e. $p(\mathbf{w}|\mathbf{X}, \mathbf{A}, h) \approx \delta(\mathbf{w} - \hat{\mathbf{w}})$ in which $\hat{\mathbf{w}} = \text{argmax}\ p(\mathbf{w}|\mathbf{X}, \mathbf{A}, h)$. The model output (1) is approximated by the degenerate distribution $\hat{p}(\mathbf{y}|\mathbf{x}, \mathbf{X}, \mathbf{A}) \approx p(\mathbf{y}|\mathbf{x}, \hat{\mathbf{w}})$. The softmax output layer predicts

a pixel-wise class probability distribution $p(\mathbf{y}_{(i,j)}|\mathbf{x}, \mathbf{X}, \mathbf{A})$. As no co-variance or dependencies between pixel-wise estimates are available, segmentation masks sampled from the pixel-wise probability distributions are often noisy [11]. An alternative approach followed by our implementation is the thresholding of pixel-wise probability values, which leads to a single, coherent segmentation map.

Ensemble methods combine multiple models to obtain better predictive performance than that obtained by the constituent models alone, while also allowing the sampling of distinct segmentation maps from the ensemble. We combine M U-Net models $h(\mathbf{x}, \mathbf{w}^{(m)})$ where, if labels from multiple annotators are available, each constituent model is trained on a disjoint label set $\mathbf{A}^{(m)}$. When trained on datasets with a single label, all constituent models are trained on the same data and their differences stem from randomized initialization and training. Treating the models as samples, we obtain an empirical distribution approximating (1) by drawing from the constituent models at random.

Monte-Carlo Dropout [4] is a Bayesian approximation technique based on dropout, where samples from the posterior over dropout weights give a better approximation of the true posterior than a MAP estimation. Given a selected model h, one can approximate (1) as $\hat{p}(\mathbf{y}|\mathbf{x}, \mathbf{X}, \mathbf{A}) \approx 1/R \sum_{r=1}^{R} p(\mathbf{y}|\mathbf{x}, \mathbf{w}^{(r)})$ when $\mathbf{w}^{(r)} \sim p(\mathbf{w}|\mathbf{X}, \mathbf{A}, h)$. Since $p(\mathbf{w}|\mathbf{X}, \mathbf{A}, h)$ is intractable, it is approximated [4] by a variational distribution $p(\theta)$ as $\theta_i = w_i \cdot z_i, z_i \sim \text{Bernoulli}(p_i)$, where p_i is the probability of keeping the weight w_i in a standard dropout scheme.

The Probabilistic U-Net [6] fuses the output of a deterministic U-Net with latent samples from a conditional variational auto-encoder modelling the variation over multiple annotators. Test-time segmentations are formed by sampling a latent $\mathbf{z}$, which is propagated with the image through the U-Net. Predictions are made as $\hat{p}(\mathbf{y}|\mathbf{x}, \mathbf{X}, \mathbf{A}) \approx p(\mathbf{y}|\mathbf{x}, \mathbf{z}^{(i)}, \hat{\mathbf{w}})$, with $\mathbf{z}^{(i)} \sim p_{\text{prior}}(\mathbf{z}|\mathbf{x})$.

4 Experiments

4.1 Data

Practical applications of uncertainty in segmentation tasks differ both in the type of ambiguity, and the availability of expert annotations. We select two representative datasets for our evaluation.

The **ISIC18** dataset consists of skin lesion images with a single annotation available [3, 15], and is used as an example of unambiguous image segmentation. We rescale the images to 256×256 pixels and split the dataset into 1500 samples for the train-set and 547 each for the validation and test sets.

The **LIDC-IDRI** lung cancer dataset [1, 2] contains 1018 lung CT scans from 1010 patients. For each scan, 4 radiologists (out of 12) annotated abnormal lesions. Anonymized annotations were shown to the other annotators, who were

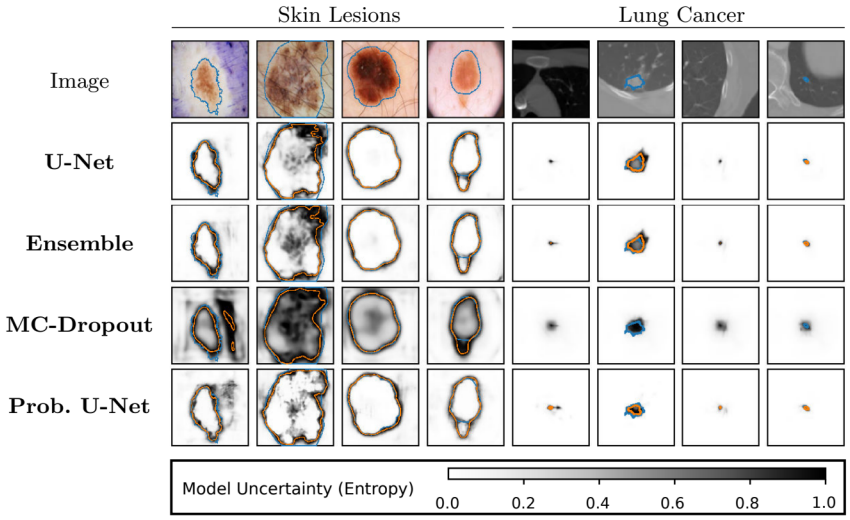


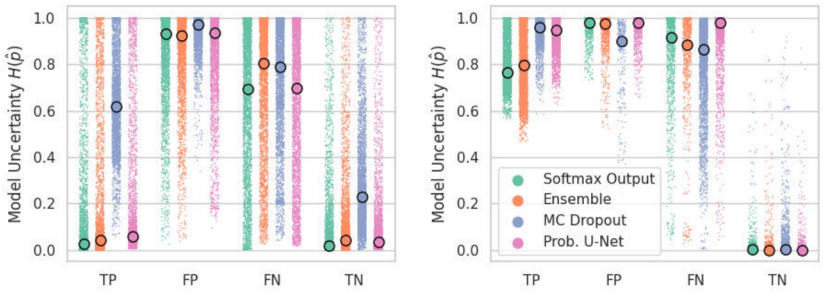
Fig. 3: Segmentation Uncertainty. Samples from the test set of the two datasets. Images in row one, model uncertainty (entropy) heat-maps in rows 2-5. Outline of mean ground truth annotations in Blue, mean model predictions in Orange.

allowed to adjust their own masks. Significant disagreement remains between the annotators: Among the extracted patches where at least one annotator marked a lesion, an average of 50% of the annotations are blank. We pre-processed the images as in [6], resampled to 0.5mm resolution, and cropped the CT-slices with lesions present to 128×128 pixels. The dataset is split patient-wise into three groups, 722 for the training-set and 144 each for the validation and test sets.

4.2 Model tuning and training

To allow for a fair evaluation, we use the same U-Net backbone of four encoder and decoder blocks for all models. Each block contains an up-/down-sampling layer, three convolution layers, and a residual skip-connection. The ensemble consists of four identical U-Nets. The latent-space encoders of the probabilistic U-Net are similar to the encoding branch of the U-Nets, and we choose a six-dimensional latent space size, following the original paper's recommendation.

All models were trained with binary cross-entropy. The probabilistic U-Net has an additional β-weighted KL-divergence loss to align the prior and posterior distributions, as per [6]. The optimization algorithm was Adam, with a learning rate of 10^{-4} for most models, except the probabilistic U-Net and MC-Dropout models on the skin lesion dataset, where a lower learning rate of 10^{-5} gave better results. We utilized early stopping to prevent over-fitting, and define the stopping criteria as 10 epochs without improvement of the validation loss, 100 epochs for models trained with the reduced learning rate. For the MC-Dropout



(a) Skin Lesion Dataset (b) Lung Cancer Dataset

Fig. 4: Pixelwise uncertainty by prediction correctness (True Positive, False Positive, False Negative, True Negative). The scatter plot shows individual pixels, with the median circled. For the lung cancer dataset, we discarded pixels with annotator disagreement.

and probabilistic U-Net models we performed a hyper-parameter search over the dropout probability p and the loss function weighting factor β, selecting the configuration with the lowest generalized energy distance on the validation set. We arrived at $p = 0.5$, $\beta = 0.0005$.

4.3 Uncertainty Estimation

For all models, our uncertainty estimates are based on non-thresholded pixel-wise predictions. For the U-Net, we take the final softmax predictions; for the remaining models we average across 16 non-thresholded samples. We quantify the *pixel-wise* uncertainty of the model by the entropy

$$\mathrm{H}(p(\mathbf{y}_{(i,j)}|\mathbf{x}, \mathbf{X}, \mathbf{A})) = \sum_{c \in C} p(\mathbf{y}_{(i,j)} = c|\mathbf{x}, \mathbf{X}, \mathbf{A}) \log_2 \frac{1}{p(\mathbf{y}_{(i,j)} = c|\mathbf{x}, \mathbf{X}, \mathbf{A})}$$

with $p(\mathbf{y}_{(i,j)} = c|\mathbf{x})$ as the pixel-wise probability to predict class $c \in C$. We plot the resulting uncertainty map for random images $\mathbf{x}$ from both datasets in Fig. 3. For visual reference, we overlay the mean expert annotation in Blue, and the mean model prediction in Orange. Darker shades indicate higher uncertainty.

We quantitatively assess the quality of uncertainty estimates by examining their relation to segmentation error in Fig. 4. On both datasets, models are more certain when they are correct (true positive, true negative) compared to when they are incorrect (false positive, false negative). A repeated measure correlation test finds a significant ($\alpha = 0.01$) correlation between segmentation error and model uncertainty on both datasets, for all methods. The relation holds, but is less strong, for MC-dropout on the skin dataset, which retains high uncertainty

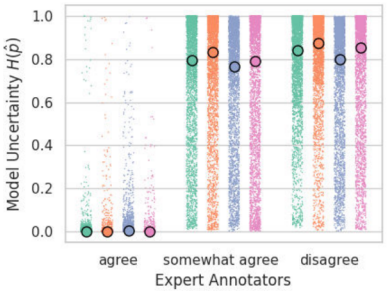


Fig. 5: Pixel-wise model uncertainty on the lung cancer dataset, grouped by agreement of expert annotations. Experts agree: $H(p) = 0$, somewhat agree $0 < H(p) < 1$, disagree $H(p) = 1$.

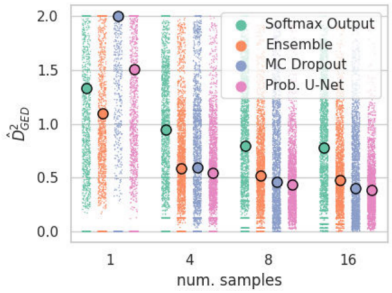


Fig. 6: Generalized Energy Distance of models on the lung cancer dataset, approximation by 1 to 16 samples, median highlighted. Lower distances are better.

even when it is correct. On the lung cancer dataset, all models have high uncertainty on true positive predictions. This might be caused by the imbalance of the dataset, where the positive class is strongly outweighed by the background and annotators often disagree. We tried training the models with a class-occurrence weighted loss function, which did produce true positive predictions with higher certainty but suffered an overall higher segmentation error.

We assess the correlation of model uncertainty with the uncertainty of the annotators on the lung cancer dataset in Fig. 5. For all models, this correlation is significant ($\alpha = 0.01$) . The median model uncertainty is very low (< 0.1) when all annotators agree, but high (> 0.7) when they disagree. There is a minor difference in model uncertainty between partial agreement (annotators split 3 – 1) and full disagreement (annotators split 2 – 2).

4.4 Sampling Segmentation Masks

Fig. 7 shows segmentation masks $\mathbf{y}$ sampled from the trained models $\hat{p}(\mathbf{y}|\mathbf{x})$. The U-Net model is fully deterministic and does not offer any variation in samples. The sample diversity of the ensemble is limited by the number of constituent models (four in our experiment). The MC-Dropout and probabilistic U-Net allow fully random sampling and achieve a visually higher diversity. On the skin lesion dataset, where only one export annotation per image is available, models still produce diverse predictions. On the lung cancer dataset, samples from the MC-Dropout and probabilistic U-Net represent the annotator distribution well.

We measure the distance between the model distribution $\hat{p}(\mathbf{y}|\mathbf{x}, \mathbf{X}, \mathbf{A})$ and the annotator distribution $p(\mathbf{y}|\mathbf{x})$ with the Generalized Energy Distance [6, 11, 14]. The distance measure is calculated as

$$D^2_{GED}(p, \hat{p}) = 2\mathbb{E}_{y\sim p, \hat{y}\sim\hat{p}}\left[d(y, \hat{y})\right] - \mathbb{E}_{y,y'\sim p}\left[d(y, y')\right] - \mathbb{E}_{\hat{y},\hat{y}'\sim\hat{p}}\left[d(\hat{y}, \hat{y}')\right] \quad . \quad (3)$$

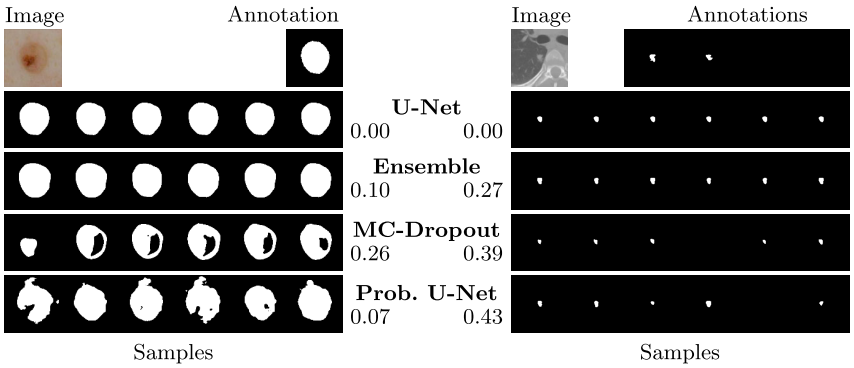


Fig. 7: Samples from the probabilistic models. First row: Image and ground truth annotations from the skin dataset (left) and lung nodule dataset (right). Following rows: samples $\mathbf{y} \sim \hat{p}(\mathbf{y}|\mathbf{x}, \mathbf{X}, \mathbf{A})$ drawn from the various models. Sample diversity over the entire dataset shown next to the model name.

We use $1 - \text{IoU}(\cdot, \cdot)$ as the distance d. A low D^2_{GED} indicates similar distributions of segmentations. We approximate the metric by drawing up to 16 samples from both distributions, and sample with replacement. The results are shown in Fig. 6. We observe that the annotator distribution is best approximated by the probabilistic U-Net, with MC-dropout and Ensemble closely behind; these pairwise ranks are significant ($\alpha = 0.01$) with left-tailed t-tests. A deterministic U-Net architecture is not able to reproduce the output distribution. Our results are consistent with [6], verifying our implementation. Following [11], we use the last term of (3) to assess the diversity of samples drawn from the model and note them in Fig. 7. They reinforce the qualitative observations of sample diversity.

4.5 Uncertainty estimates for active learning

Instead of training the models with all available data $\{\mathbf{X}, \mathbf{A}\}$, we now start with a small random subset $\{\mathbf{X}_0, \mathbf{A}_0\}$. We train the model with this subset at iteration $t = 0$, and then add a set of k images from $\{\mathbf{X}, \mathbf{A}\}$ to form $\{\mathbf{X}_{t+1}, \mathbf{A}_{t+1}\}$. Samples are selected based on the sum of pixel-wise entropies [7]. We repeat for T iterations, benchmarking against a random sample selection strategy.

For both skin lesion and lung cancer datasets, we start with a training size of 50 images, add $k = 25$ images at each iteration, and repeat $T = 10$ times. The models are trained for 5000 gradient updates with a batch size of 16 and 32 for the respective datasets. Since annotations are costly and to speed up computations, no validation-loss based early stopping is used. The experimental setup has been picked to ensure meaningful model uncertainties for the data selection policy and to ensure convergence within each active learning iteration.

The learning curves in Fig. 8 show that random-based sampling leads to a faster reduction in test loss over the uncertainty-based sampling strategy for

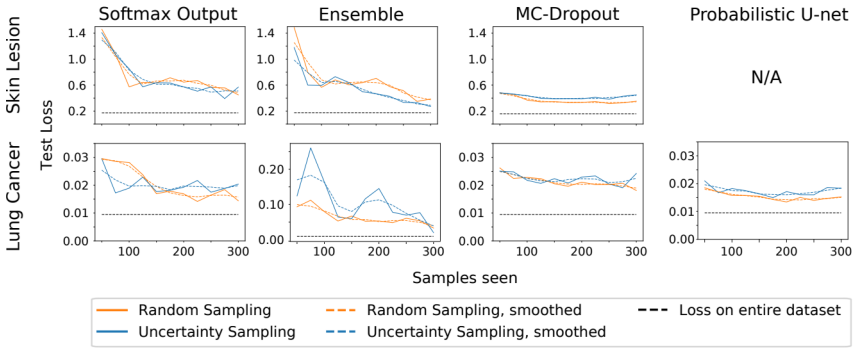


Fig. 8: Learning curves for the four algorithms on both datasets. Note that the Probabilisitic U-net only applies to the ambiguous segmentation.

Fig. 9: An example of the ambiguous samples frequently selected for inclusion into the training set under the uncertainty-based data gathering strategy. This unseen sample was selected when 150 annotations were revealed. The group of expert annotators provided disagreeing segmentation masks, confirming the model uncertainty but providing little additional information to learn from.

both datasets. We further investigated the samples selected by the uncertainty-based strategy by looking at the images which caused a large increase in the test error. One such image is shown in Fig. 9.

5 Discussion & Conclusion

Our results in Fig. 4 show that there is a clear relation between uncertainty estimates and segmentation error. The examples in Fig. 3 further highlight that areas of high uncertainty are not merely distributed around class boundaries, but also encompass areas with ambiguous labels. Fig. 5 shows that the uncertainty estimates obtained from the model are a good representation of the uncertainty of a group of expert annotators. We conclude that pixel-wise model uncertainty estimates give the practitioner a good indication of possible errors in the presented segmentation mask, allowing those predictions to be examined with care.

The learning curves in Fig. 8 show that estimated uncertainty is not generally useful for selecting active learning samples, for any model or dataset. Our results

depend on using the sum of pixel-wise entropies as a per-image entropy, which is correct for the softmax model, but only an approximation for the other models. This might impact our results. For the Lung Cancer dataset, all models estimate high uncertainty for the positive class, and the active learner thus selects images with a large foreground, skewing the proportion of classes represented in the training set. Furthermore, the selected images often have high annotator disagreement, illustrated in Fig. 9. If the active learner prefers sampling ambiguous images, it will be presented with inconsistent labels leading to harder learning conditions and poor generalisation. This may stem from an incorrect active learning assumption that annotations are noise-free and unambiguous, which is often not true. In conclusion, for a fixed budget of annotated images, we find no advantage in uncertainty-based active learning.

We observed similar behaviour of pixel-wise uncertainty estimates across all four segmentation models. The models differ in their ability to generate a distribution of distinct and coherent segmentation masks, with only the MC-dropout and probabilistic U-Net offering near unlimited diversity (see Fig. 7). But these models are harder to implement, more resource-intensive to train, and require hyperparameter tuning. The choice of model is ultimately application dependent, but our experiments show that even a simple U-net is competitive for the common task of assessing segmentation error. This agrees with [5], which compared uncertainty quantification models for unambiguous segmentation.

Our division of segmentation tasks into ambiguous and unambiguous considers it as "unambiguous" when a fundamentally ambiguous segmentation task is covered by a single annotator – or potentially several annotators, but with only one annotator per image, as for the Skin Lesion dataset. Even if the underlying task *is* ambiguous, the models considered in this paper inherently assume that it is *not*, as there is no mechanism to detect annotator variance when every image is only annotated once. More fundamental modelling of segmentation ambiguity and uncertainty thus remains a highly relevant open problem.

To conclude – is segmentation uncertainty useful? We find that uncertainty, even in the simplest models, reliably gives practitioners an indication of areas of an image that might be ambiguous, or wrongly segmented. Using uncertainty estimates to reduce the annotation load has proven challenging, with no significant advantage over a random strategy.

Acknowledgements. Our data was extracted from the "ISIC 2018: Skin Lesion Analysis Towards Melanoma Detection" grand challenge datasets [3, 15]. The authors acknowledge the National Cancer Institute and the Foundation for the National Institutes of Health, and their critical role in the creation of the free publicly available LIDC/IDRI Database used here. This work was funded in part by the Novo Nordisk Foundation (grants no. NNF20OC0062606 and NNF17OC0028360) and the Lundbeck Foundation (grant no. R218-2016-883).

References

1. Armato III, S.G., McLennan, G., Bidaut, L., McNitt-Gray, M.F., Meyer, C.R., Reeves, A.P., Zhao, B., Aberle, D.R., Henschke, C.I., Hoffman, E.A., et al.: The lung image database consortium (LIDC) and image database resource initiative (IDRI): a completed reference database of lung nodules on CT scans. Medical physics **38**(2), 915–931 (2011)
2. Clark, K., et al.: The cancer imaging archive (TCIA): maintaining and operating a public information repository. Journal of digital imaging **26**(6), 1045–1057 (2013)
3. Codella, N.C., Gutman, D., Celebi, M.E., Helba, B., Marchetti, M.A., Dusza, S.W., Kalloo, A., Liopyris, K., Mishra, N., Kittler, H., et al.: Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (ISBI), hosted by the international skin imaging collaboration (ISIC). In: 2018 IEEE 15th International Symposium on Biomedical Imaging. pp. 168–172
4. Gal, Y., Ghahramani, Z.: Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In: international conference on machine learning. pp. 1050–1059 (2016)
5. Jungo, A., Reyes, M.: Assessing reliability and challenges of uncertainty estimations for medical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 48–56. Springer (2019)
6. Kohl, S., Romera-Paredes, B., Meyer, C., De Fauw, J., Ledsam, J.R., Maier-Hein, K., Eslami, S., Jimenez Rezende, D., Ronneberger, O.: A probabilistic u-net for segmentation of ambiguous images. Advances in neural information processing systems **31**, 6965–6975 (2018)
7. Lewis, D.D., Catlett, J.: Heterogeneous uncertainty sampling for supervised learning. In: Machine learning proceedings 1994, pp. 148–156. Elsevier (1994)
8. Loog, M., Yang, Y.: An empirical investigation into the inconsistency of sequential active learning. In: 2016 23rd international conference on pattern recognition (ICPR). pp. 210–215. IEEE (2016)
9. MacKay, D.J.: The evidence framework applied to classification networks. Neural computation **4**(5), 720–736 (1992)
10. MacKay, D.J.: Information-based objective functions for active data selection. Neural computation **4**(4), 590–604 (1992)
11. Monteiro, M., Le Folgoc, L., Coelho de Castro, D., Pawlowski, N., Marques, B., Kamnitsas, K., van der Wilk, M., Glocker, B.: Stochastic segmentation networks: Modelling spatially correlated aleatoric uncertainty. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 12756–12767 (2020)
12. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
13. Settles, B.: Active learning literature survey. Tech. rep., University of Wisconsin-Madison Department of Computer Sciences (2009)
14. Székely, G.J., Rizzo, M.L.: Energy statistics: A class of statistics based on distances. Journal of statistical planning and inference **143**(8), 1249–1272 (2013)
15. Tschandl, P., Rosendahl, C., Kittler, H.: The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Scientific data **5**, 180161 (2018)
16. Yang, Y., Loog, M.: A benchmark and comparison of active learning for logistic regression. Pattern Recognition **83**, 401–415 (2018)

Appendix C

U-net segmentation of tubular structures from live imaging confocal microscopy: Successes and challenges

The following chapter presents the extended abstract

> Kasra Arnavaz, Pia Nyeng, Jelena M. Krivokapic, Oswin Krause, Aasa Feragen. "U-net segmentation of tubular structures from live imaging confocal microscopy: Successes and challenges" Virtual Early Career European Microscopy Congress (EMC) 2020.

This work presents the approach we took to segmenting the tubular structure in pancreas using a U-net [32] architecture. It can be considered as a prelude to the Appendix D. We make a case for why this data set is particularly difficult to segment: Firstly, there is a low signal-to-noise ratio, which is an inevitable property of live imaging data since laser power has to be kept low to prevent killing the cells. Moreover, due to biological variations between embryos, movies are quite different in their signal strength and intensity distributions. This challenge is further heightened by limited annotations, as our training set consists of 6 3D images which are drawn from 3 movies, while we have recorded more than 50 movies.

We then mention our solutions to the challenges posed above. Since the training set is small compared to the huge diversity of movies we would like our model to generalize to, we preprocess each image by its own mean and variance rather those of the training set. This resulted in a significant boost in segmentation of those movies with a low signal-to-noise ratio. To get a more accurate estimate of our performance while keeping the annotation labor to a minimum, we evaluate our performance on smaller patches which are drawn from several movies rather than full-sized images drawn from a few movies. On a different note, our evaluation metrics are pixel-wise while biologists are more interested in having an accurate topology. We conclude that a semi-supervised approach which takes topology into account could be a promising future work, which we pursued in Appendix D.

U-net segmentation of tubular structures from live imaging confocal microscopy: Successes and challenges

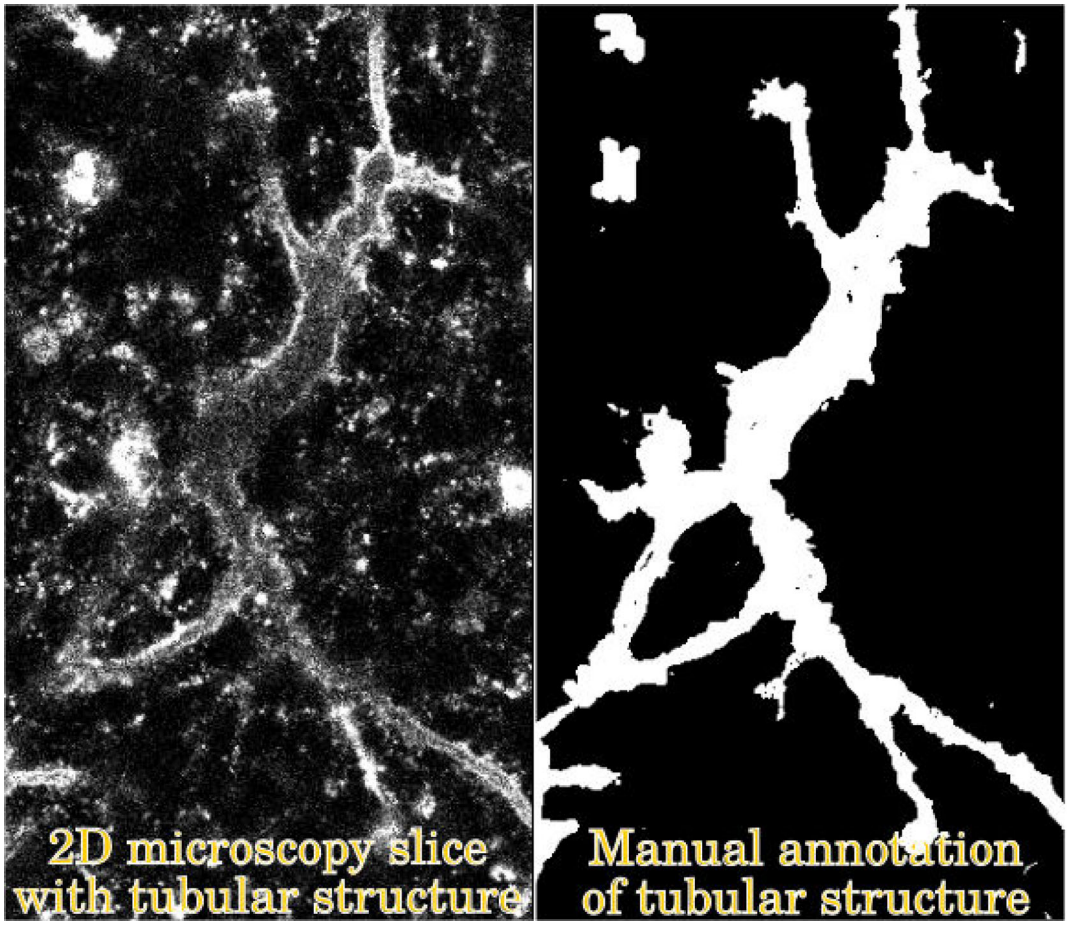
Kasra Arnavaz, Pia Nyeng, Jelena Miskovic Krivokapic, Oswin Krause, Aasa Feragen

Summary

We study the problem of segmenting tubular structures in the pancreas from live imaging via confocal microscopy. The segmentation is difficult due to noise, making it challenging even for a human expert to distinguish structure from background. We present the results of the state-of-the-art U-Net and inspect its successes and failures. We discuss challenges in the manual segmentation of the structure, the impact of these challenges on our choice of experimental design, and future directions for improving the segmentation.

Introduction

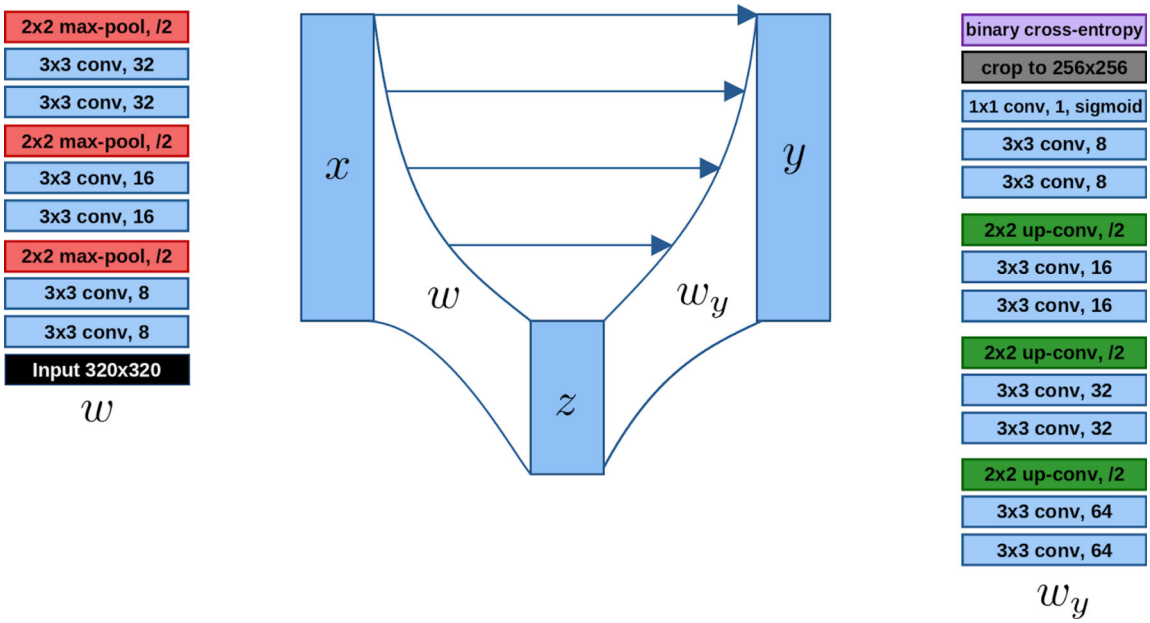
During embryonic development, the pancreatic tubes (Figure 1) grow from a web-like network with loops to a tree-like structure without loops. A thorough understanding of the mechanisms governing this morphology has yet defied developmental biologists (Shih et al. 2013), but observations indicate that the topological events happening during development of the tubular structure may be related to differentiation of beta cells nearby. However, confirming this quantitatively requires a binary segmentation of the tubes.



While segmentation of other tubular structures such as airways or vessels is well studied, the pancreatic tubes are different in the sense that we have no prior anatomical knowledge about their structure. Moreover, the images are recorded with low signal to noise ratio, creating a challenging segmentation task. We present results of the state-of-the-art U-net (Ronneberger et al, 2015) and discuss its successes and failures. In particular, we discuss challenges in the manual labeling of training data, as well as potential strategies for "optimally" labeling data.

Methods/Materials

We implement the U-net (Figure 2) in Keras, using 3 sets of down- and up-sampling layers and binary cross entropy loss, and train it using the Adam Optimizer (beta_1 = 0.9, beta_2 = 0.999, lr = 10^{-5}) for 50 epochs.



Our dataset consists of 2D image slices extracted from 3D image stacks obtained from live imaging using confocal microscopy of the developing pancreas of mouse embryos imaged for 40-48 hours in vitro (see abstract submission from P Nyeng).

The 2D slices are aggregated as follows: A total of 6 3D movies with approximately 220 time-points each are split randomly into two sets with 3 movies each, to be used for training and testing, respectively. As parts of the movies have very high noise levels with no discernible structure, a smaller set of images is manually subselected by first randomly sampling, from each movie, two 3D images at random. These images have a resolution of 1024 x 1024 x D, where D varies between 27 and 40. Each of these images is divided into a 4 x 4 grid of patches in the x-y plane, each of size 256 x 256 x D. From this set of patches, a trained laboratory assistant manually removes all patches which are too dark and noisy to be reliably annotated. The remaining patches typically form connected parts of the x-y plane, leading to 1024 x 1024 x D images with ``forbidden regions" consisting of the removed patches.

Manual annotation: The training split consists of 6 3D images from 3 different movies with in total 174 2D image slices, where the tubular structure is fully annotated in a semi-manial fashion as follows: The 3D images were first processed in Imaris image processing software using Z-normalization, smoothing, volume rendering and filtering of the Muc1-mcherry signal. The resulting segmentations are manually edited by the laboratory assistant using the software Slicer. To validate on as diverse a dataset as possible, we select from the test split 10 3D patches at random, in which the tubular structure is annotated by the laboratory assistant, giving annotations for around 13\% of the test split.

Image preprocessing: While the mean intensity per movie is close to zero, the variance differs noticeably. Thus, every time-point is stadardized by its own mean and standard deviation, and intensities are clipped to be at most 3 standard deviations away from the mean (which is zero).

Training on patches: The U-net is trained on image patches from the fully annotated slices in the training set, avoiding the forbidden regions.

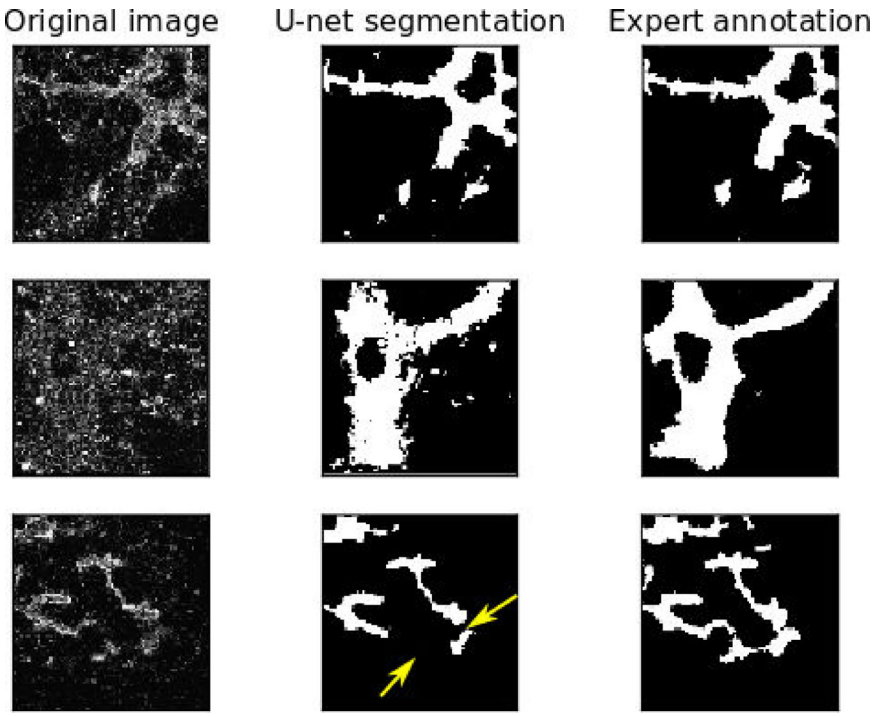
Results and Discussion

On the test set, the U-net segmentation produces:

- a sensitivity, giving the proportion of true foreground segmented pixels out of ground truth foreground pixels, of 0.648;
- a specificity/true negative rate giving the proportion of true background segmented pixels out of ground truth background pixels, of 0.987
- An accuracy, giving the proportion of correctly segmented pixels over all pixels, of 0.970
- A dice score, giving a similarity between the predicted segmentation and the ground truth segmentation, of 0.687 (optimal score is 1).

Note that the accuracy, which is often used to assess quality of segmentation, is high due to the large imbalance between foreground and background, where much of the dominant background class is easy to predict. As shown by the sensitivity and specificity, however, the U-net is undersegmenting the foreground.

This is problematic for two reasons. First, the expert annotation is already likely an undersegmentation, due to the fact that the expert annotator is unlikely to annotate foreground unless she is fairly confident. Second, undersegmentation easily leads to disruptions in the topology of the segmented tubular network, as a few missing pixels can be enough to cut a connected tube into two or more pieces.



Conclusion

We have demonstrated how a U-net trained on patches from a limited set of 2D slices from a limited set of 4D microscopy live images obtains a qualitatively reasonable performance. In light of the limited amount of training data, with limited amount of variation, the results are positive, although they also present challenges: The U-net is shown quantitatively to undersegment the tubular structure, which raises some concern regarding the correctness of segmented network topology.

In theory, we expect that this could be handled by increasing the amount of hand-annotated training data. In practice, however, manual labeling is expensive, difficult and time consuming. This thus leads to multiple future research directions, including semisupervised learning to take advantage of our large amounts of unlabeled data, and including spatial and time information into the segmentation pipeline. Finally, we note that the segmentation problem is callenging, also for a human, who is able (more so than a neural network) to take contextual information into account. A final potential research problem is thus to specialize the network to search for tubular structures.

References

HP Shih, A Wang, M Sander, Annu Rev Cell Dev Biol 29 (2013), p.81-105
O Ronneberger, P Fischer, T Brox, MICCAI (2015), p.234-241

Appendix D

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

---

The following chapter presents the article

> Kasra Arnavaz, Oswin Krause, Kilian Zepf, Jakob Andreas Bærentzen, Jelena M. Krivokapic, Silja Heilmann, Pia Nyeng, and Aasa Feragen. "Quantifying Topology In Pancreatic Tubular Networks From Live Imaging 3D Microscopy" under review at Machine Learning for Biomedical Imaging (MELBA) 2021.

This work is a continuation of what was presented in Appendix C. It proposes solutions to the problems that were encountered in using a U-net segmentation model to segment pancreatic tubes.

In this paper, we derive a topological score function which measures topological similarity between a ground truth- and a predicted segmentation. This score function is well-suited for the purposes of the biological application we have in this thesis, but it can in principle be used for any topological segmentation problem. It is highly interpretable as it is composed of a component- and a loop subscore, which can each be further broken down to analogues of recall and precision. The proposed score function can be used for evaluation as well as model selection.

A recurring challenge throughout this thesis has been the cost of annotations. To this end, we compare the performance of three architectures: 1. a fully supervised U-net, 2. AE+U-net: a U-net whose encoder has been pretrained on an autoencoder, and 3. semisupervised U-net: a joint training of a U-net and an autoencoder with a shared encoder. Our experiments indicate a better generalization for the semisupervised U-net compared to the fully-supervised U-net.

Lastly, accurate detection and tracking of loops in the tubular structure of the pancreas is a key factor for testing the biological hypothesis. We use the temporal information in recorded movies as a post-processing step to filter out short-lived loops. Doing so increases the loop score significantly.

Journal of Machine Learning for Biomedical Imaging. In submission

1

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

2 Motivated by the challenging segmentation task of pancreatic tubular networks, this paper
3 tackles two commonly encountered problems in biomedical imaging: Topological consis-
4 tency of the segmentation, and expensive or difficult annotation. We propose a topological
5 score which measures both topological and geometric consistency between the predicted
6 and ground truth segmentations, applied to model selection and validation. We use a
7 semisupervised U-net architecture, applicable to generic segmentation tasks, which jointly
8 trains an autoencoder and a segmentation network. This allows us to utilize unannotated
9 data to learn feature representations that generalize to test data with high variability, in
10 spite of our annotated training data having very limited variation. We then use tracking of
11 loops over time to further improve the predicted topology. Our contributions are validated
12 on a challenging segmentation task, locating tubular structures in the fetal pancreas from
13 noisy live imaging confocal microscopy.

14 **Keywords:** topology, semisupervised, segmentation, tubular, confocal microscopy

15 1. Introduction

16 Segmentation of tubular structures is a common task in medical imaging, encountered when
17 analyzing structures such as e.g. blood vessels (Zhang et al. (2019)), airways (Qin et al.
18 (2019)), ductal systems (Wang et al. (2020b)), neurons (Li et al. (2019)). Segmentation is

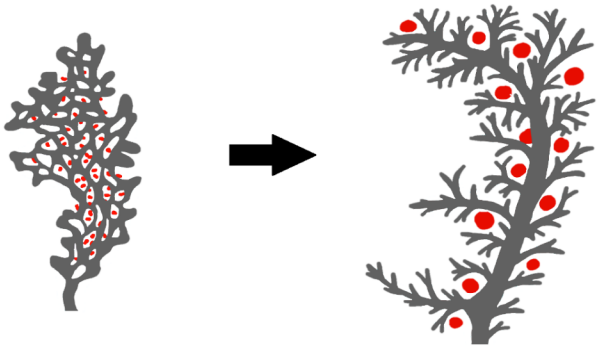


Figure 1: Tubular remodeling in pancreas during embryonic development. Gray areas represent tubes, while red dots represent β-cells.

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.

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,

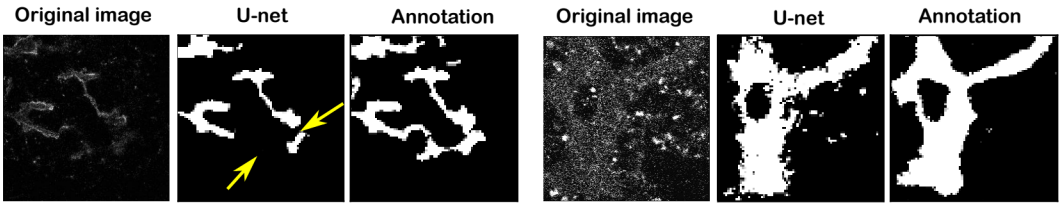


Figure 2: 2D slices. **Left:** Small voxel error, large network error. **Right:** Effect of noise and artifacts.

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 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: At the time when this paper was developed, current versions Hu et al. (2019) applied to 2D (or 2.5D) images, where they are forced to resort to enforcing topological consistency of small patches rather than globally. 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. Second, computational topology tends 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 in Ouali et al. (2020) take a different approach to incorporating unlabeled data to learn better hidden representation features. They first train an encoder and a decoder on labeled data, and then add in auxiliary decoders in addition to the original decoder for unlabeled data. The idea is to improve the shared encoder by ensuring auxiliary decoder outputs to be consistent with the original decoder output. 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.

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. This semisupervised U-net has a similar structure to the 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 common encoder network. While they apply this approach to labeled data, we make use of the unlabeled data as well as the labeled data to learn better latent space representations.

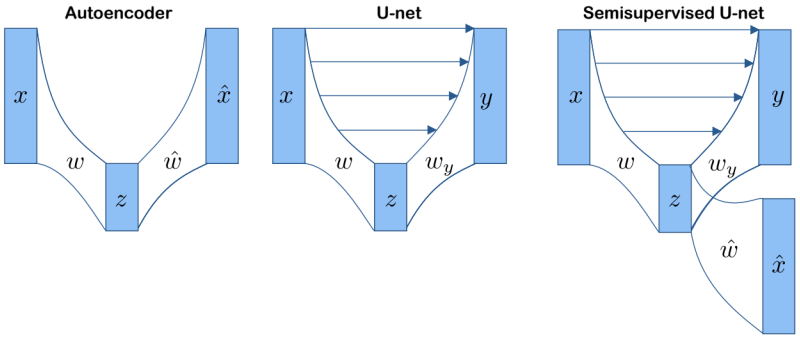


Figure 3: We apply semisupervised training of a segmentation U-net by combining an autoencoder (left) with a U-net (middle) into a combined network (right) which is simultaneously trained for segmentation and reconstruction. This allows us to learn features on a large, unlabeled dataset with rich variation, 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.

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 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.

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}. 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.

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. 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

$$\frac{\mathrm{TP}_{gt} + \mathrm{TP}_p}{\mathrm{TP}_{gt} + \mathrm{TP}_p + 2\mathrm{FP} + 2\mathrm{FN}}.$$

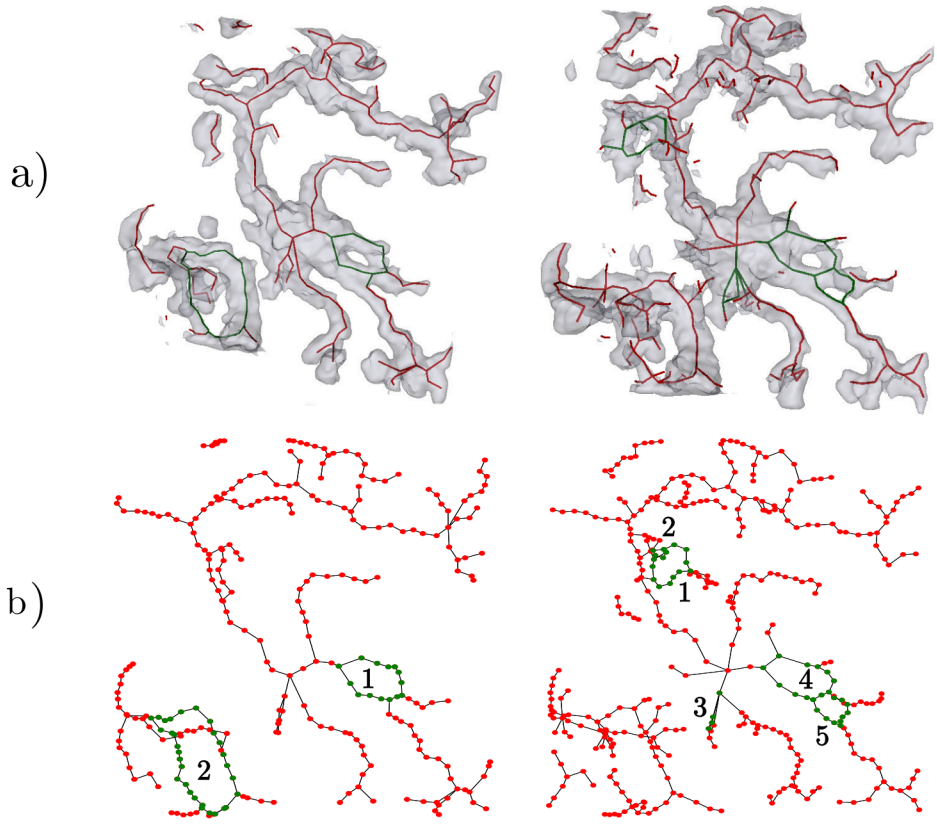


Figure 4: Ground truth (left) and prediction (right): a) segmentation and their skeletons. b) 2D projections of the skeletons onto the (x, y)-plane shown as nodes and edges. Green lines in a) and green nodes in b) take part in loops. Note that the ground truth segmentation has 2 loops and 7 components, while the prediction has 5 loops and 10 components. The loops are identified by numbers, for the sake of the discussion below.

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:

$$\mathrm{IoU} = \frac{\dfrac{\mathrm{TP}_{gt}}{\mathrm{TP}_{gt} + \mathrm{FN}} + \dfrac{\mathrm{TP}_{p}}{\mathrm{TP}_{p} + \mathrm{FP}}}{\dfrac{\mathrm{TP}_{gt} + 2\mathrm{FN}}{\mathrm{TP}_{gt} + \mathrm{FN}} + \dfrac{\mathrm{TP}_{p} + 2\mathrm{FP}}{\mathrm{TP}_{p} + \mathrm{FP}}}, \tag{1}$$

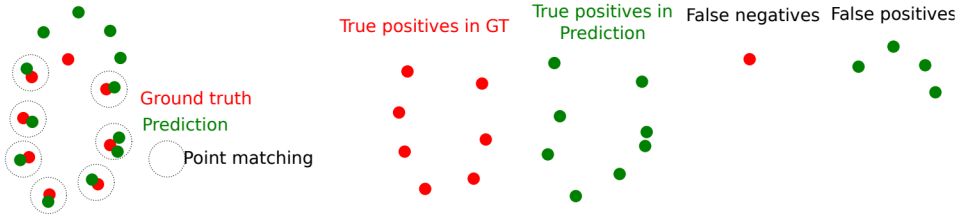


Figure 5: An example of point matching for two loops (edges not drawn).

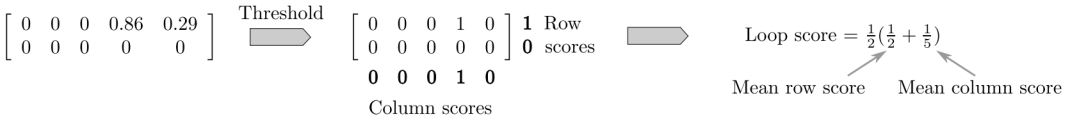


Figure 6: Example computation of the loop score for the prediction shown in Figure 4. The ground truth segmentation has 2 loops while the prediction has 5. Here, loop #1 in S_{gt} has overlap with loops #4 and #5 in S_p, where the first is thresholded as a perfect match and the second is thresholded 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.

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

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)

By adding horizontally we obtain a row score indicating the degree to which a ground truth loop is matched to a loop in predicted topology. Similarly, summing vertically, the column score indicates the degree to which a predicted loop is matched to a loop in 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.

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). We chose to annotate a small number of full 3D images in order to have some manually segmented examples in which a complete tubular network was visible.

For the validation- and test sets, we chose to annotate patches rather than full images to obtain maximal data variation. This was prioritized both to select robust models and to obtain a robust assessment of performance. The validation- and test sets, each compose of 17 different movies. The patches to be segmented were drawn randomly from those that were not labeled as low information patch. Following this, our validation- and test sets

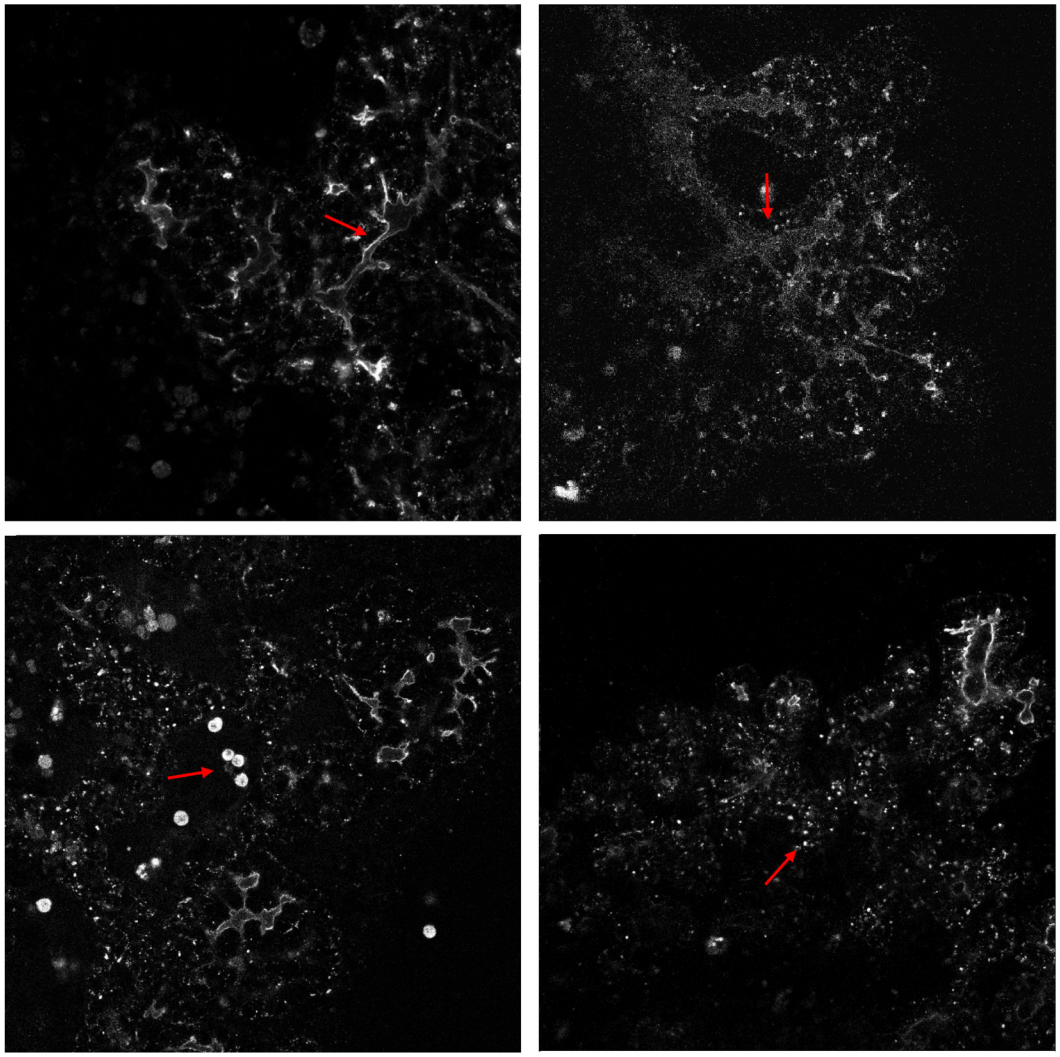


Figure 7: 2D slices from different movies extracted to show different image artifacts. **Top left:** a case where the boundary of tube has a high intensity. **Top right:** a case where the intensity of boundaries are no higher than the rest of the tube. **Bottom left:** large bright spots due to red blood cells and other dead cells. **Bottom right:** small bright spots due to the reporter protein in cells being transported to the tubular surface.

consisted of 35 and 68 manually segmented 3D image patches, respectively. Each test set patch was also given a labeling difficulty score, from 0 = easy to 3 = difficult.

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 patch. 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. 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 in the combined loss of the semisupervised U-net was set to $\alpha = 10$. 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.

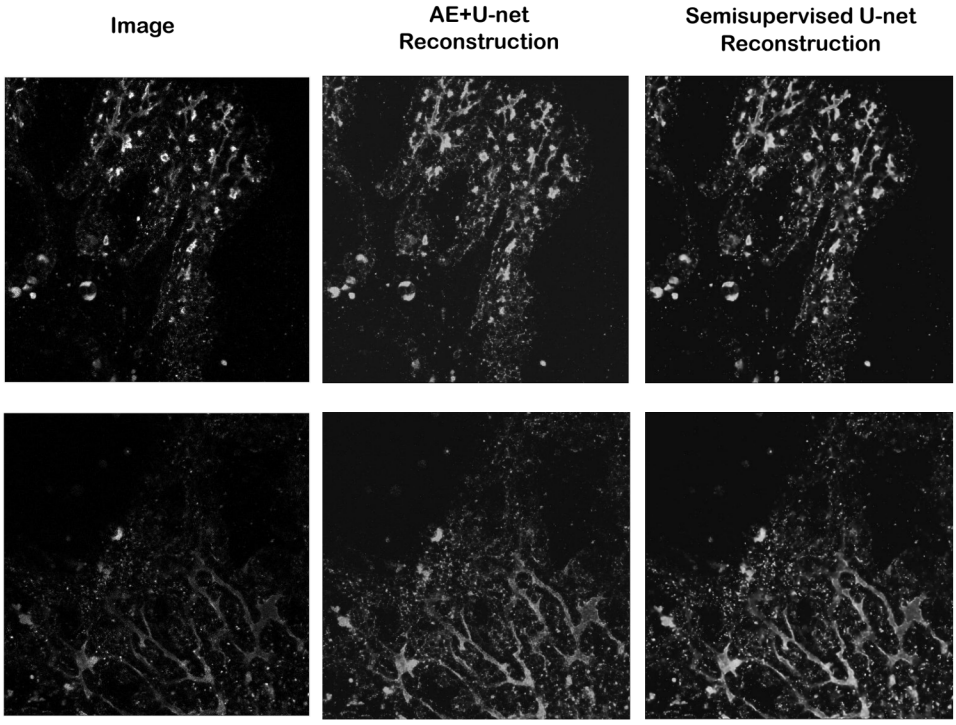


Figure 8: Two 2-dimensional slices from the test set (left column) and their corresponding reconstruction at the final epochs of AE+U-Net (middle column), and semi-supervised U-net (right column).

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 criterion 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
U-net	Voxel IoU	0.8	0.474±0.208	0.535±0.192	0.395±0.263	0.312±0.297	0.477±0.285
	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
AE+U-net	Voxel IoU	0.5	0.447±0.210	0.504±0.196	0.356±0.243	0.308±0.299	0.404±0.264
	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
Semisupervised U-net	Voxel IoU	0.7	**0.490±0.226**	**0.555±0.213**	**0.436±0.253**	0.386±0.352	**0.486±0.271**
	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			0.449±0.193	0.443±0.187	0.337±0.263	0.314±0.346	0.360±0.275

5.2 Loop filtering and tracking

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, namely the semisupervised U-net at its best-performing threshold of 0.7.

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 (search range = 15), (adaptive stop = 5), (adaptive step = 0.9), (memory = 1), and (threshold = 15). 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**.

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 2 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.

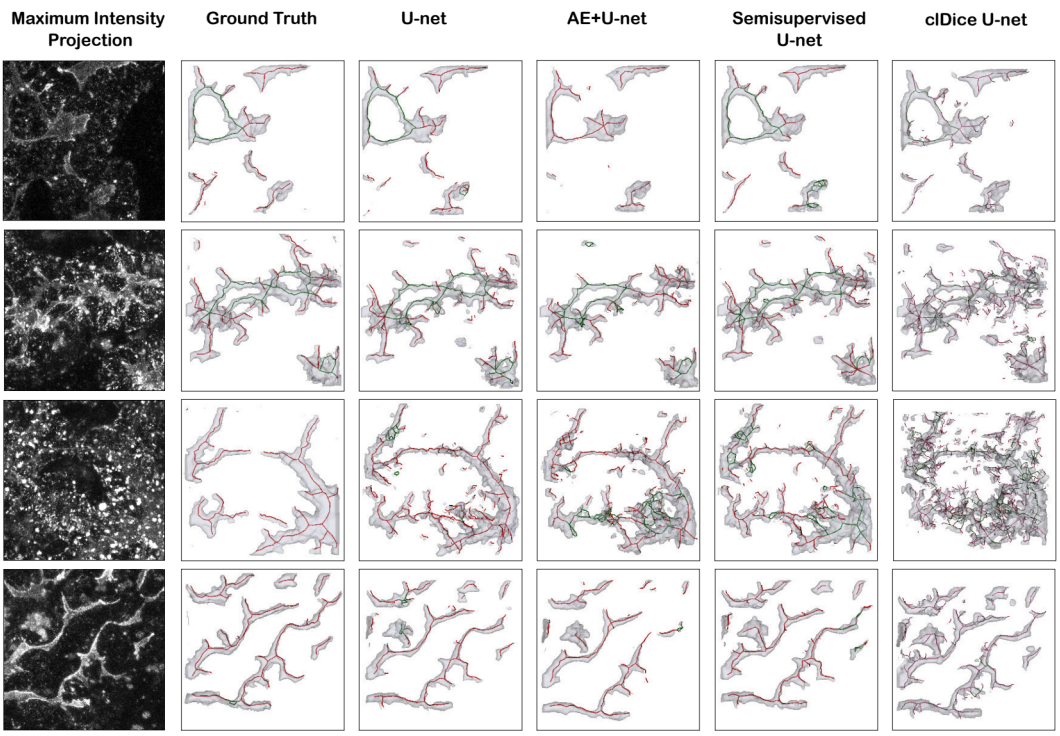


Figure 9: Segmentation results for 4 example test patches.

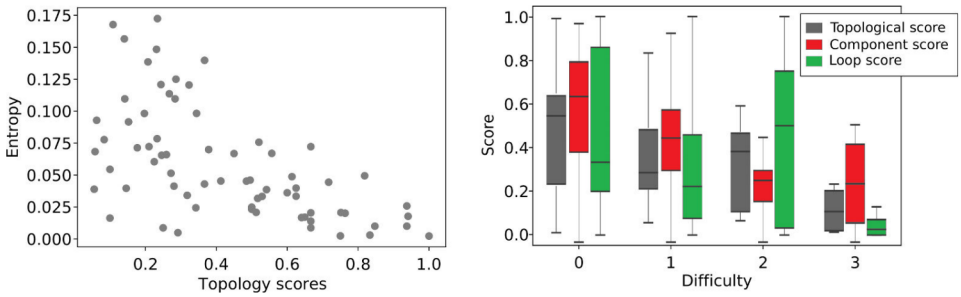


Figure 10: **Left**: Patch-wise topology score versus segmentation entropy. **Right**: Distribution of topology, component and loop scores divided over difficulty of manual labeling.

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

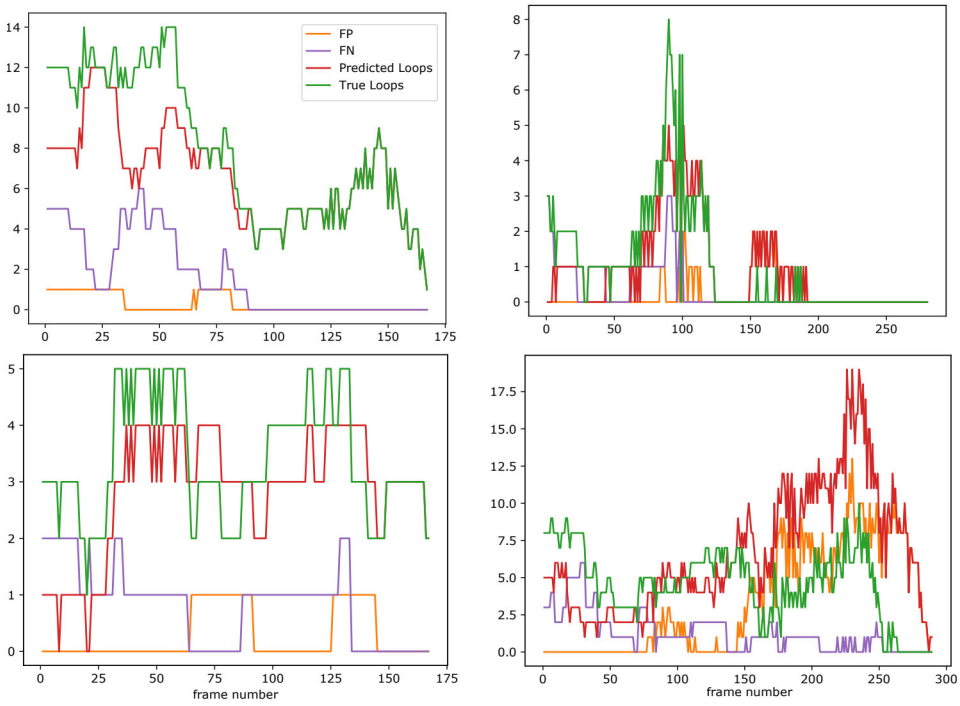


Figure 11: Each plot is a movie from the test set showing the performance of loop detection evaluated by an expert. The y-axis measures the number of false positives (FP), false negatives (FN), predicted loops, and true loops.

Table 2: Performance of tracking loop centers for the semisupervised U-net architecture.

HOTA	DetA	DetRe	DetPr	AssA	AssRe	AssPr
0.337	0.256	0.361	0.649	0.469	0.478	0.956

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.

A topological score function We have derived a topological score function from highly interpretable constituents for both loops and components, which split into subscores anal-

ogous 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 2 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.

Our segmentation problem is challenging, and in solving 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.

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.

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.

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.

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.