



Technical University of Munich

School of Computation, Information and Technology - Informatics

Master's Thesis in Biomedical Computing

Segmentation of sparse annotated data: application to cardiac imaging

Joshua Stein



Technical University of Munich

School of Computation, Information and Technology - Informatics

Master's Thesis in Biomedical Computing


**Segmentation of sparse annotated data:
application to cardiac imaging**

**Segmentierung spärlich annotierter Daten:
Anwendung auf die kardiale Bildgebung**

Author:	Joshua Stein
Supervisor:	Prof. Julia Schnabel
Advisor:	Dr. Maxime Di Folco
Submission Date:	October 15th 2023

I confirm that this master's thesis is my own work and I have documented all sources and material used.

Submission date: October 16th 2023

Signature: 

Abstract

Short-axis cardiac magnetic resonance imaging (cMRI) is an important clinical task. Segmentation of cMRI images into cardiac structures, such as the ventricles and myocardium, can yield important information for diagnosis, treatment and management of disease. Training state-of-the-art neural networks for segmentation requires large volumes of annotated data. Unfortunately, acquiring such annotated data is expensive and time-consuming. As a result, much work has focused on using fewer data (e.g. transfer learning, data augmentation, contrastive learning, etc.). To the best of our knowledge, none of this work has investigated which particular types of sparsity are most important for training performant networks for the task of correctly segmenting the ventricles and myocardium. There has also been limited work investigating fine-tuning foundation models for cMRI segmentation. In this work we investigate how training with sparse data (i.e. reducing the number of cases annotated), training with sparse annotations (i.e. reducing the number of slices annotated per case) and training with sparse input prompts (in the form of points and bounding boxes) affect performance. We evaluate segmentation performance on the state-of-the-art segmentation models nnU-Net and SAM on two public datasets. We show that using a significantly reduced dataset (48 annotated volumes) can result in models that achieve Dice scores of 0.85, producing segmentations that are comparable to training with all available data (160 and 240 volumes for each dataset respectively). Similar Dice scores can be achieved when training on 10 randomly sampled slices per volume (this corresponds to using approximately 50% and 70% of available slices on either dataset respectively). We further show that training using mid-ventricular slices yield the best performing networks, and training using apical slices the worst. In general, annotating more slices per volume is a better strategy than annotating more volumes with fewer slices. Finally, we show that baseline foundation model performance is limited, achieving Dice scores of up to 0.70 with correct prompting (using bounding boxes, with both positive and negative sample points). This score can be improved to 0.84 and 0.76 on either dataset using appropriate fine-tuning.

Contents

Abstract	v
List of Figures	ix
List of Tables	xi
1 Introduction	1
2 Background	3
2.1 Anatomy and physiology [14]	3
2.2 Cardiac Magnetic Resonance Imaging (cMRI)	4
2.3 U-Net	5
3 Related Work	7
4 Method	9
4.1 Segmentation networks	9
4.1.1 nnU-Net	9
4.1.2 Segment Anything Model (SAM)	9
4.2 Sparsity	11
4.2.1 Sparsity of volumes	11
4.2.2 Sparsity of slices	11
4.2.3 Annotation strategy	12
4.3 Evaluation metrics	12
4.3.1 Dice score	12
4.3.2 Hausdorff distance	13
4.3.3 Mean absolute distance	13
5 Experiments and results	15
5.1 Datasets	15
5.2 Sparse data	15
5.3 Sparse annotations	17
5.4 Sparse data vs sparse annotations	19
5.5 Segment Anything Model (SAM)	21
6 Discussion	29
6.1 Sparse data	29
6.2 Sparse annotations	30
6.3 Sparse data vs sparse annotations	31

6.4 Segment Anything Model (SAM)	32
6.4.1 Baseline inference performance	32
6.4.2 Fine-tuning performance	33
7 Conclusion	35
Acknowledgments	37
Bibliography	39

List of Figures

2.1	The four chambers of the heart, along with major arteries and veins [15].	3
2.2	The cardiac planes [8].	4
2.3	Examples of short-axis cMRI images. From top to bottom are apical slices, mid-ventricular slices and basal slices. Images are sourced from the ACDC dataset [7].	5
2.4	The U-Net architecture [17]. White boxes represent copied feature maps.	6
4.1	When unprompted, baseline SAM will often over-segment an image.	10
4.2	Iteratively reducing the amount of training data from a fully annotated dataset to only a single volume fully annotated.	11
4.3	Iteratively training on only particular cardiac regions (zeroed out regions shown in grey).	12
4.4	Dice score	13
5.1	Example cMRI images, with slice annotations. From top to bottom are an apical slice, a mid-ventricular slice and a basal slice. As we move from the basal to the apical region the size of the cardiac structures decrease. RV = right ventricle, MYO = myocardium and LV = left ventricle.	16
5.2	Qualitative difference between 3D nnU-Nets trained on one tenth of available volumes. The top row shows a slice from the ACDC dataset, predicted by a network trained on 16 volumes (with 20 slices per volume, for a total of 320 slices). The bottom row show a slice from the M&Ms dataset, predicted by a network trained on 24 volumes (with 14 slices per volume, for a total of 336 slices).	17
5.3	Qualitative results of segmentations produced by networks trained on different cardiac regions on the ACDC dataset. From top to bottom are slices from the basal region, the mid-ventricular region and the apical region respectively. Each slice is extracted from a different 3D segmentation volume. GT = ground truth. AMB means the network was trained on apical/middle-ventricular/basal slices (and permutations thereof).	18
5.4	Resulting spurious segmentation for a 3D nnU-Net trained only on mid-ventricular slices on the ACDC dataset.	18
5.5	Qualitative results demonstrating improved performance as the network is trained on an increased number of annotated slices/decreased number of volumes. The first row is a random slice from a 3D segmentation on the ACDC dataset. The second row is the same on the M&Ms dataset. V=volumes, S=slices and represents the total number of volumes/slices each network was trained on respectively.	20

5.6	Segmentation results on the right ventricle, generated by baseline SAM on the M&Ms dataset. From top to bottom the model is prompted with only positive points, positive and negative points, positive points with a bounding box and finally positive points, negative points and a bounding box. Positive and negative samples are shown as green and red stars respectively.	22
5.7	Effect of using negative samples when running inference on the myocardium. The top row shows sampling only using a bounding box and two positive myocardium samples (shown as green stars). We note the resulting over-segmentation. The bottom row shows the same, but with a negative sample from each of the ventricles (red stars); while not perfect, the segmentation is much more accurate.	23
5.8	Qualitative results showing the difference in outputs of SAM models fine-tuned on the ACDC dataset. Models were fine-tuned with bounding boxes, using 2 positive points and 1 negative point per class.	24
5.9	Qualitative results showing the difference in outputs of SAM models fine-tuned on the M&Ms dataset. Models were fine-tuned with bounding boxes, using 2 positive points and 1 negative point per class.	26
5.10	Qualitative results showing the difference in outputs of fine-tuned SAM models prompted without bounding boxes , using 2 positive points and 1 negative point per class. Models were trained with all available data. . .	27

List of Tables

5.1	Effect of training different nnU-Nets on sparse annotated volumes. Note that the ACDC dataset only has 160 volumes.	16
5.2	Effect of training a 3D nnU-Net with sparsely annotated cardiac regions (A=apical slices, M=middle slices, B=basal slices). The network is evaluated on all regions.	18
5.3	Influence of training with randomly sampled and sparsely annotated slices from all three cardiac regions using 3D nnU-Net. Note that there are only 14 slices per volume (after pre-processing) for the M&Ms dataset.	19
5.4	Segmentation performance of a 3D nnU-Net when changing the number of training volumes (V) vs the number of slices (S) while keeping the total number of slices approximately 1400. Note that the M&Ms dataset has a total of 14 slices per volume, and that the ACDC dataset has a total of 160 volumes.	20
5.5	Influence of keeping slices constant while reducing number of training volumes. Trained on ACDC with a 3D nnU-Net network.	20
5.6	Influence of keeping slices constant while reducing number of training volumes. Trained on M&Ms with a 3D nnU-Net network.	21
5.7	Baseline SAM inference results on the ACDC dataset. Positive and negative sample counts are per each segmentation class. All Dice standard deviations are less than 0.15. Unless shown, HD standard deviations are less than 5mm and MAD standard deviations are less than 2mm.	24
5.8	Baseline SAM inference results on the M&Ms dataset. Positive and negative sample counts are per each segmentation class. All Dice standard deviations are less than 0.15. Unless shown, HD standard deviations are less than 5mm and MAD standard deviations are less than 2mm.	25
5.9	Inference results for SAM models fine-tuned with limited training data. The models were prompted with bounding boxes , two positive sample points and one negative sample per class. Dice standard deviations, HD standard deviations and MAD standard deviations are less than 0.1, 3mm and 1.2mm for all models respectively. Note that the ACDC dataset only has 160 volumes.	25
5.10	Inference results of a SAM model, fine-tuned on the ACDC dataset. The models were prompted without bounding boxes , two positive sample points and one negative sample per class. Dice standard deviations are less than 0.1 for all models. Unless shown, MAD standard deviations are less than 2mm.	26

5.11 Inference results of a SAM model, fine-tuned on the M&Ms dataset. The models were prompted without bounding boxes , two positive sample points and one negative sample per class. Dice standard deviations are less than 0.1 for all models. Unless shown, MAD standard deviations are less than 1.8mm for all models respectively.	26
---	----

1 Introduction

Cardiovascular diseases (CVDs) are the leading causes of global mortality, causing about 30% of all deaths in 2019 [1]. They are also a leading contributor to disability - the World Health Organization (WHO) estimates that ischaemic heart disease and stroke were the second and third biggest contributors to disability adjusted life years (DALYs) in 2019 [2]. Alarming, both the prevalence of CVDs and the trend for DALYs have steadily increased since 1990 [3]. These burdens affect all populations, with over 75% of CVD-caused deaths taking place in low and middle income countries [4]. Diagnosing and treating CVDs is therefore a global priority.

Different non-invasive imaging techniques such as magnetic resonance imaging (MRI), computed tomography and ultrasound are used for CVD diagnosis, treatment planning, monitoring and prognosis. Each of these modalities have their own advantages and disadvantages, and could be suitable depending on the patient case. However, cardiac magnetic resonance imaging (cMRI) has become the gold standard for cardiac imaging, due to the high soft tissue contrast it presents. As a first step towards diagnosis of CVD, a cardiac image may be segmented into different anatomical regions. Physicians are primarily interested in segmenting the left ventricle (LV), right ventricle (RV) and myocardium (MYO). These can yield the most important quantitative diagnostic information, such as ventricular ejection fractions, ventricular stroke volumes and myocardium thickness [5–8].

The medical computer vision community has, for many years, had interest in segmentation of cardiac images [9]. Over the past several years, with the rise of deep learning and accessible compute power, deep learning has become the state-of-the-art method of choice for segmentation [10]. Alongside the rise in deep learning popularity, several important datasets and challenges have been published. Popular examples include the Sunnybrook Cardiac Data (SCD) [11], the Left Ventricle Segmentation Challenge (LVSC) [12], the Automatic Cardiac Diagnosis Challenge (ACDC) [7] and the M&Ms Challenge [13].

Deep learning methods require large datasets to learn from. A significant difficulty in assembling these datasets is the cost of annotation. Medical images are very expensive to annotate - they require expert domain knowledge, and fine-grained segmentations need to be drawn, often between soft tissue boundaries that do not have clear delineations. This is apparent when looking at the size of available datasets, and the number of annotations within those datasets. For example, SCD has only 45 cMRIs. More recent datasets from technical challenges still remain limited (the ACDC and M&Ms challenges have only 150 and 375 cMRIs respectively).

As a result of this, many methods have arisen that focus on trying to learn networks using limited data.

In general, the data used in these methods are *sparse* or *limited*. There are different definitions in the literature for what exactly sparsity means. We understand sparsity to be in the data, in the annotations or in the form of the annotations themselves.

Sparsity of data means that the dataset itself is small. In general, most medical datasets are sparse in data due to the aforementioned high cost of annotation. Here, in the context of cMRI, we consider sparsity in data to mean only having a few MRI volumes.

Sparsity of annotations means that there are a limited number of annotations for each data point. When considering cMRI volumes, a volume that has only a few slices annotated would be considered sparse.

Sparsity in annotations themselves refers to the types of annotations provided, and the density of information therein. Some datasets provide pixel-level semantic segmentations. Other datasets, however, may only provide points, bounding boxes or scribbles.

Despite the high cost of annotation and the significant amount of work done in investigating how best to take advantage of sparse data, very little work has been done investigating exactly what type of sparsity is most important. For example, should we favour data sparsity over annotation sparsity (that is, prefer many volumes with only a few slices annotated, or only a few volumes with many slices annotated)? Or if we prefer annotation sparsity, are there particular annotations or slices that are more important than others? Are there particular cardiac regions (apical, mid-ventricular or basal) that are more important to annotate than others?

We aim to investigate these questions by analysing the performance changes in state-of-the-art segmentation networks when trained on different sparse inputs. For example, we can compare segmentation performance of networks trained with sparse data compared to those trained on sparse annotations.

We also aim to investigate the role of foundation models specialised for segmentation. Foundation models have good baseline performance on a variety of upstream tasks and can be fine-tuned to achieve strong performance on novel tasks. They are also promptable - we are able to provide inputs in the form of sparse points or bounding boxes. We aim to better understand the role that these sparse annotations have on performance, and determine if fine-tuned foundation models can offer similar performance to networks trained from scratch.

By analysing the performance of networks trained with different sparse inputs we aim to better understand the role of sparse data on performance, and further to determine the boundaries at which good segmentation results are possible.

2 Background

2.1 Anatomy and physiology [14]

The heart is the primary organ of our cardiovascular system. It is divided into two sides, left and right and further into four chambers - two atria, at the top of the heart and two ventricles beneath them. Each side of the heart has one atrium and one ventricle, connected by valves. The heart is broad at the superficial surface, the base, and tapers towards the apex.

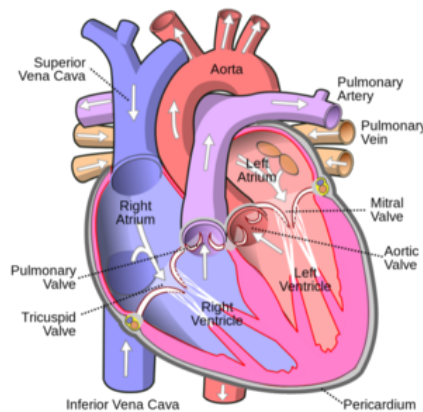


Figure 2.1: The four chambers of the heart, along with major arteries and veins [15].

From a physiological perspective the heart can be considered as being split into two separate circuits. The systemic circuit is responsible for pumping oxygenated blood to the entire body and returning de-oxygenated blood to the heart. The pulmonary circuit is responsible for pumping de-oxygenated blood the lungs and returning oxygenated blood. De-oxygenated blood becomes oxygenated as a result of gaseous exchange between the air in our lungs and our capillaries.

The heart is composed of three layers - the superficial epicardium, the middle myocardium and the deep pericardium. The thickest layer is the myocardium, composed of cardiac muscle cells. This thick muscular layer is required for the continuous pumping of blood throughout our body.

The heart beats through a well-defined cycle, consisting of two phases. During systole the heart muscles contract, pumping blood into circulation. During diastole the muscles relax, allowing blood to fill the chambers.

The cycle begins with *atrial systole*, during which the atria contract, forcing blood into the ventricles. Next begins *ventricular systole*. At the start of ventricular systole the

ventricles are filled - the volume of blood in the ventricles is the end diastolic volume (EDV). After filling, the ventricles contract, pumping blood away from the heart. The amount of blood remaining in the ventricles after this ejection phase is the end systolic volume (ESV). The difference between EDV and ESV is the amount of blood pumped to the body and is known as stroke volume. *Atrial diastole* occurs simultaneously with ventricular systole, during which the atria fill with blood. Finally, *ventricular diastole* occurs, during which the ventricles relax, and the atria begin to contract again, starting a new cycle.

2.2 Cardiac Magnetic Resonance Imaging (cMRI)

cMRI is the gold standard for assessing cardiac structure and function. It provides excellent in-plane resolution and high soft-tissue contrast. A cine sequence is used, wherein a series of images are taken repeatedly over an area and then played back as a sequence [16].

Imaging can be taken along the short-axis (transverse) plane, the long-axis (sagittal) plane or the four-chamber (coronal) plane. These planes are relative to the heart, not the body. The short-axis plane gives an excellent cross-sectional view of the ventricles and is frequently used to assess EDV and ESV.

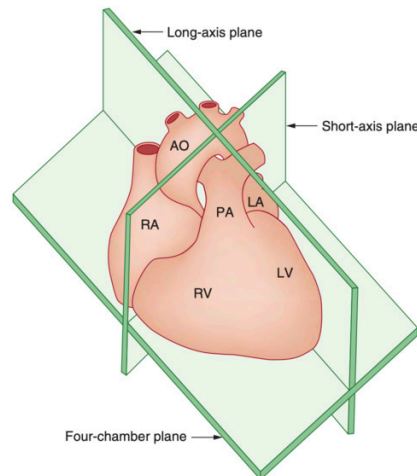


Figure 2.2: The cardiac planes [8].

cMRI enables ‘(1) assessments of myocardial perfusion in the investigation of ischaemic heart disease; (2) differentiation of the aetiology of non-ischaemic cardio-myopathies; and (3) characterisation of congenital heart syndromes pericardial disease, and cardiac masses’ [8]. Frequently, segmentation of the different cardiac chambers plays an important role in these use cases. Example short-axis cMRI slices are shown in Figure 2.3.

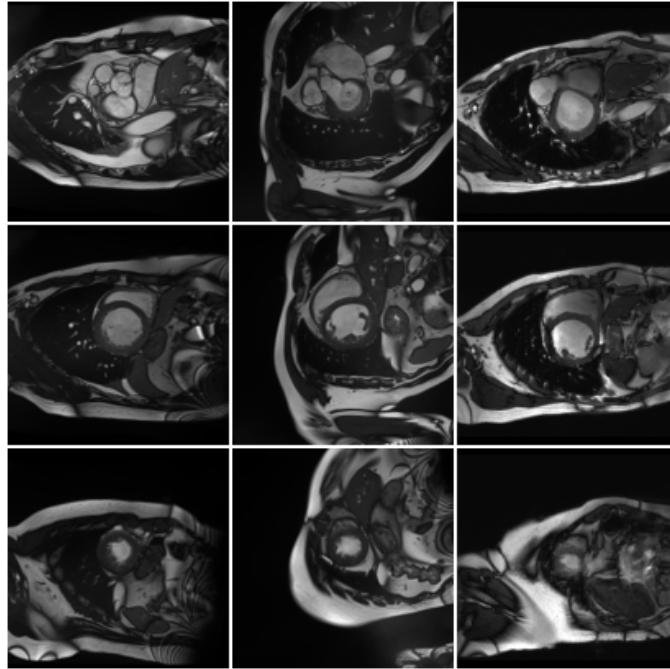


Figure 2.3: Examples of short-axis cMRI images. From top to bottom are apical slices, mid-ventricular slices and basal slices. Images are sourced from the ACDC dataset [7].

2.3 U-Net

U-Net [17] is an encoder-decoder network designed for medical image segmentation. It has two paths - a downwards contracting path, which learns an image encoding (thereby capturing image context), and an upwards expanding path, which learns an output segmentation from an image encoding (enabling precise localisation). Throughout the contracting path, a series of convolutions, ReLU and max-pooling are applied. This results in a small (in terms of height and width) but deep (in terms of number of channels) encoding. In the expanding path upsampling, followed by convolutions are applied. The resulting upsampled feature map is concatenated with the corresponding cropped feature map from the contracting path. The concatenated feature map is again convolved and passed through a ReLU activation. The final layer of the network is a 1x1 convolution that maps from the 64 channels of the final feature map to the number of classification classes. The architecture of U-Net is shown in Figure 2.4

The original paper uses strong data augmentation, needed due to a limited number of training samples. The authors consider random elastic deformations as particularly important, but also use random shifts, rotations and grey-scale perturbations. This challenge of limited training data remains a persistent problem in medical image segmentation.

The original U-Net won the 2015 ISBI cell tracking challenge. Over the past several years, many newer approaches have been built on the base encoder-decoder architecture of the original network [18]–[21]. These variations have continued to win an array of medical challenges, continuously producing state-of-the-art results [22]. As a result of its superb

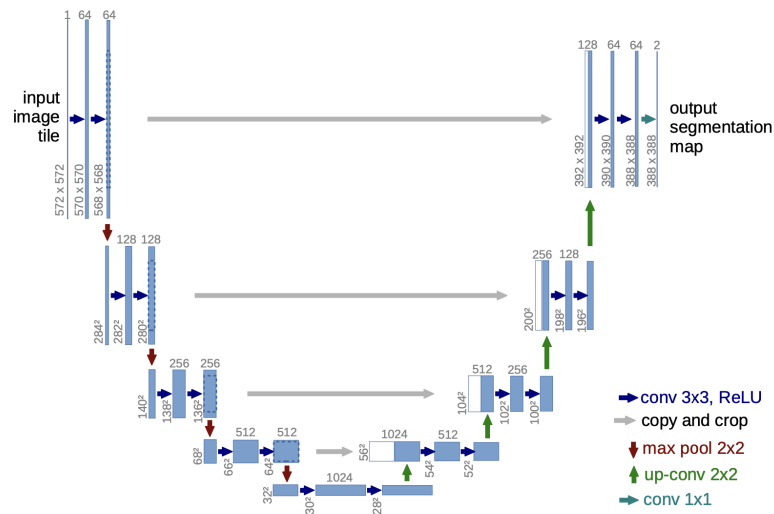


Figure 2.4: The U-Net architecture [17]. White boxes represent copied feature maps.

performance, U-Net has become one of the most important contributions to medical image segmentation. Although originally applied to microscopic images for cell segmentation, subsequent work has used U-Net for cMRI segmentation. This is discussed further in Chapter 3.

3 Related Work

Segmentation is a broad topic, and there is a large body of work focused on semantic segmentation. Here, we examine the most important related works, with a focus on sparse cMRI segmentation. Some general methods for tackling sparsity include transfer learning, data-augmentation, semi-supervised learning and self-supervised learning - many others exist [23].

Bai et al. [24] propose semi-supervised learning wherein unlabelled data is utilized in the training process. They alternate between performing segmentation on unlabelled data and updating model parameters (using true labels and estimated labels). They evaluate their model on short-axis cMRI images from the UK biobank [25]. The semi-supervised models outperformed baseline supervised models and multi-atlas segmentation models.

Bai et al. [26] combine convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to incorporate both spatial and temporal information for aortic segmentation. They address the key challenge of sparse annotations by performing label propagation using non-rigid image registration [27] to estimate the motion between successive frames. New pseudo-labels are generated by propagating existing labels using the learned registration transformation. The propagated labels are down-weighted in the training loss to account for accumulated error. The authors outperform a baseline UNet [17] as measured using Dice score, contour distance and temporal smoothness.

Similarly Bitarafan et al. [28] also develop an approach using registration and label propagation. However, they use a 2D network to determine inter-slice registration and use only a single 2D slice as the source to propagate. They are able to out-perform other baseline networks, and achieve an average Dice score within 5% of a fully supervised UNet model.

Significant work has also gone into exploring strategies that leverage contrastive learning. Chaitanya et al. [29] use a combination of global and local features during the contrastive learning phase. Global features are learned similarly to normal contrasting features [30]. Here, similarity refers to slices from within the same partition of a volume, or slices within the same partition across different volumes (depending on the contrastive learning strategy). Local features refer to features *within* an image that are similar/dissimilar. Assuming two volumes are roughly aligned, similarity means having aligned patches extracted from slices from different volumes. The authors are able to pre-train on many unlabelled data, and then fine-tune using a small dataset. On ACDC pre-training is performed on 52 volumes, followed by fine-tuning on 1, 2, or 8 volumes. Using 8 volumes, the authors achieve a Dice score within 3% of a baseline UNet (trained using 78 labelled volumes).

Zeng et al. [31] build on this work by introducing Positional Contrastive Loss (PCL). Their main contribution is to define similarity according to the position of slices within a volume. Slices close together are considered similar, and those far apart dissimilar (the exact tolerance for the definition of ‘close together’ and ‘far apart’ is determined empirically

according to the dataset). The authors show using more patients for fine-tuning saturates performance - the best results (compared to baseline) are in the extreme cases, when only very few (less than 10) patients are used for fine-tuning.

Recently, You et al. [32] introduced MONA, a contrastive learning framework for 2D medical image segmentation. It is designed to handle tailness, consistency and diversity within medical datasets. Using their contrastive learning framework, built on iterative-similarity distillation (ISD) [33] and fine-tuning with 10% labelled data, they are able to achieve comparable performance to a U-Net trained with full supervision.

Over the past few years there has been significant research interest into *foundation models* [34]. These models are trained on a broad dataset, and can later be fine-tuned for downstream tasks. Popular models include BERT [35], GPT [36] and CLIP [37]. Typically, these models require vast compute power and enormous datasets to train. Once trained, fine-tuning to downstream tasks can become a powerful way to deal with sparse data.

Earlier this year the Segment Anything Model (SAM) was released as a foundation model for segmentation [38]. SAM is a promptable segmentation model composed of three parts: (1) an image encoder to compute an image embedding, (2) a prompt encoder to compute a prompt embedding and (3) a mask decoder which takes the two embeddings as input and outputs segmentation masks. The model is trained on a partially self-generated dataset consisting of more than 1 billion segmentation masks. The model achieves impressive zero-shot performance, often out-competing fully supervised networks.

There have been several approaches to implementing SAM for medical image segmentation. Cheng et al. [39] assess SAM's zero-shot performance using a variety of prompting methods on a curated collection of 12 medical datasets. They show that box-prompts without jitter yield the best performance. Similarly, Huang et al. [40] evaluate different prompts on 52 medical image segmentation datasets. They show that without prompting SAM often performs poorly. Performance improves through a variety of different prompts (e.g. with positive points, negative points and/or bounding boxes), albeit inconsistently. Again, it is shown that box-prompts yield the best results, suggesting that the rich positional information allows SAM to perform better. The authors conclude that SAM struggles on objects with ill-defined boundaries, poor tissue contrast and small sizes.

He et al. [41] compare SAM to a variety of U-Nets on 12 different datasets. They show that all the U-Nets outperform SAM, suggesting that the baseline SAM model has limited generalisation ability on medical images.

Ma et al. [42] introduce MedSAM. They use a similar network structure to SAM, and pre-train on the SAM dataset. They then fine-tune the image encoder and mask decoder on a curated medical image dataset comprised of more than 1 million images. They show that MedSAM outperforms a baseline U-Net and baseline SAM across a variety of tasks and modalities.

4 Method

4.1 Segmentation networks

4.1.1 nnU-Net

nnU-Net (no new net) [43] is a self-configuring U-Net [17] that automatically adapts its pipeline for new segmentation tasks. Rather than creating a specialised network, loss function or training scheme, the authors leverage the principles of automatic machine learning [44] to define a system that automatically configures pre-processing, training and post-processing for any new task. The pipeline defines a ‘recipe’ for learning that uses three parameter groups: fixed parameters, rule-based parameters and empirical parameters.

Fixed parameters do not require any adaptation - they are consistent between datasets and runs. For example, the loss function (a combination of Dice and cross-entropy) and the optimiser (stochastic gradient descent with Nesterov momentum) are both fixed parameters.

Rule-based parameters are defined as heuristics that adapt to each dataset. For example, the intensity normalisation strategies changes depending on the data type. For CT images, global percentile clipping and z-score normalisation is performed with global mean and standard deviation. For other modalities, only z-score normalisation is performed with per-image mean and standard deviation.

Empirical parameters are learned entirely from each dataset. For example, choosing the best model from the learned ensemble of 2D, 3D and 3D cascaded models is a decision that requires the full dataset and assessment of cross-validation performance.

As with the original U-Net, strong data augmentation is applied. Augmentations include random rotations, scaling, Gaussian noise, blurring, brightness and contrast perturbations, simulation of low resolution, gamma correction and mirroring.

nnU-Net is the current state-of-the-art for medical image segmentation, having been shown to outperform specialised pipelines on a range of tasks with strong generalisation characteristics.

We train nnU-Net models using a variety of sparse segmentations. Each model was trained for 20 epochs. When training on particular cardiac regions (see Section 4.2.2) we disable oversampling, which would otherwise cause slices to be sampled from outside the regions of interest. All other parameters of the original nnU-Net pipeline are respected. Training code is available at https://github.com/joshestein/nnUNet/tree/limited_data.

4.1.2 Segment Anything Model (SAM)

SAM [38] is currently the largest foundation model for image segmentation. Although not explicitly trained on medical images, it has shown excellent zero-shot performance

on non-medical tasks. This naturally raises the question of how well it will perform on new medical tasks, and whether it can be used to as a baseline for fine-tuning a more performant medical model.

As outlined in Chapter 3, SAM is composed of three primary parts - an image encoder, a prompt encoder and a mask decoder. During training/inference we are able to modify the quality of inputs to the model by changing the type and number of prompts.

As a naïve base input, one is able to not provide any prompt information to SAM (this is sometimes referred to as *everything* or *auto* mode). In this mode, SAM will simply try to segment as many objects as possible in the input image. For medical images, this is often unreliable; the object of interest may be small in comparison to the rest of the image, and may not be clearly delineated. Further, there may be too much detail in a medical image for the model to ‘know’ what is the correct segmentation. An example of a segmentation mask produced that fails to segment the cardiac structures is shown in Figure 4.1.

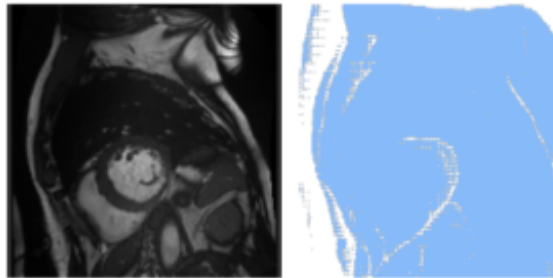


Figure 4.1: When unprompted, baseline SAM will often over-segment an image.

Alternative prompt information can be provided in the form of text input, bounding boxes and positive or negative points. For medical images, these modes are often more reliable as they allow the model to ‘focus’ on regions of interest, producing more accurate segmentation masks. This requires that all input images contain all classes of interest. If a sampled input image does not contain all foreground classes, it is not used for inference or fine-tuning. When using SAM we prefer not to use multi-mask outputs. We therefore run inference per class and subsequently combine the results into a single output segmentation mask. That is, we segment first the right ventricle, then the myocardium, then the left ventricle *independently* and then combine each of these outputs into a single output mask.

When fine-tuning, we fine-tune only the mask decoder. That is, we freeze the image encoder and prompt encoder, and propagate gradients only through the mask decoder. Since running the forward pass through the image encoder is the most expensive part of the pipeline, we dynamically save image encodings to disk when sampling new slices. These encodings are re-used during subsequent runs. A single encoding for a a single slice is approximately 4 megabytes, so the storage cost is minimal, and the read/write overhead is insignificant compared to passing an input through the image encoder. The default ViT-H model is used for all inference and fine-tuning experiments.

In general, SAM is designed for broad generalisation and depth across a variety of segmentation tasks, rather than specialised state-of-the-art performance on any single task. As noted in the original paper, SAM can ‘miss fine-structures, hallucinate small

disconnected components... and does not produce boundaries as crisply as more computationally intensive methods'. Even so, it is interesting to question how well we can adapt the baseline model for specialised cMRI segmentation. Fine-tuning code is available at https://github.com/joshestein/TUM_thesis.

4.2 Sparsity

Medical datasets are often considered sparse. In our experiments, we consider sparsity to mean sparsity of volumes, sparsity of slices or sparsity in the prompt annotation information.

4.2.1 Sparsity of volumes

We investigate the effect of training on sparse datasets by randomly sampling a number of patients (i.e. volumes) from the entire dataset. By iteratively changing our sample size, we can determine the threshold at which performance deteriorates due to too little training data.

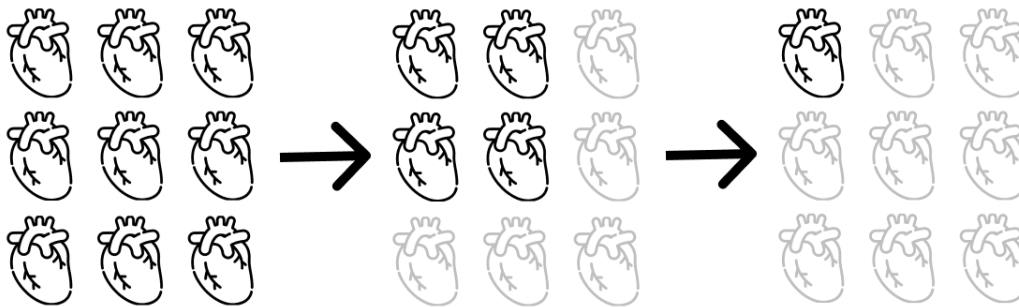


Figure 4.2: Iteratively reducing the amount of training data from a fully annotated dataset to only a single volume fully annotated.

4.2.2 Sparsity of slices

Sparse annotations are investigated by randomly zeroing out a certain number of slices from a 3D volume. By zeroing the slices, we do not alter the volume size - this prevents us from needing to modify the training pipeline.

Slices are either randomly zeroed from across the entire region, or zeroed in some particular regions. We assume that each volume can be split into thirds, with the first third containing basal slices, the second third containing mid-ventricular slices and the final third containing apical slices.

By only training on particular permutations of slice regions we can determine if some regions are more influential on performance than others. For example, if we want to investigate the performance of training only on apical slices, we can zero-out mid-ventricular

and basal slices and train on the total volume (for a 3D network) or slices only from the apical region (for a 2D network).

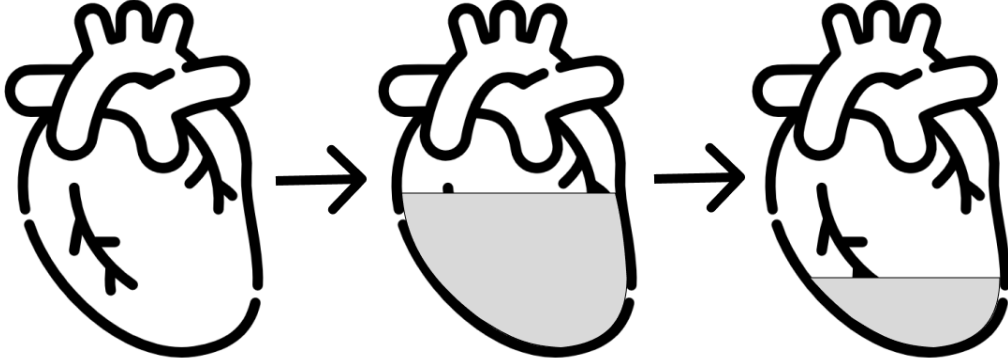


Figure 4.3: Iteratively training on only particular cardiac regions (zeroed out regions shown in grey).

4.2.3 Annotation strategy

The best annotation strategy for nnU-Nets is determined by fixing the total number of slices as constant and altering the proportion of total volumes vs total slices annotated. This allows us to determine if it would be better to annotate more volumes (with fewer slices) or more slices (with fewer volumes).

When annotating inputs to SAM, we use a variety of positively sampled points, negatively sampled points and bounding boxes. To sample points we take inspiration from Huang et al. [40]. Our initial point is sampled from the foreground class’s centre of mass. If that point is part of the foreground class, it is persisted as the initial point (otherwise it is discarded). Subsequent points are uniformly sampled at random from the remaining foreground pixels. When using bounding boxes, we simply find a box around the class of interest with a margin of 5 around the most extreme pixels.

4.3 Evaluation metrics

4.3.1 Dice score

The Dice score is a commonly used metric in segmentation tasks. It is a measure of overlap between a prediction and ground truth defined as:

$$\text{Dice} = \frac{2TP}{2TP + FP + FN}$$

where TP , FP , FN are true positive, false positive and false negative predictions respectively.

Dice can also be expressed as twice the intersection of true positive predictions over the union of the prediction with ground truth. This is shown in Figure 4.4.

The score ranges from 0 to 1, where 0 can be interpreted as having no overlap between the prediction and ground truth and 1 indicates perfect overlap.

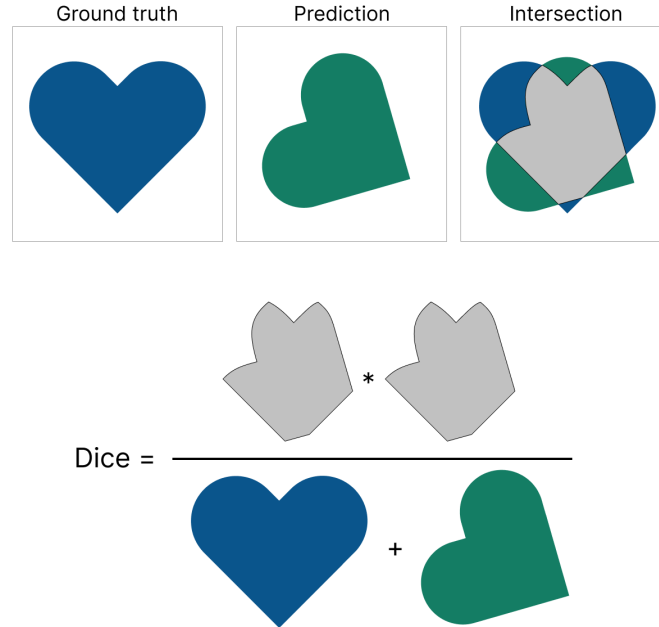


Figure 4.4: Dice score

4.3.2 Hausdorff distance

Hausdorff Distance (HD) is a surface distance metric that measures the how far two point sets are from each other. For two point sets X and Y , it is defined as:

$$\text{HD}(X, Y) = \max_{x \in X} \min_{y \in Y} \|x - y\|$$

This is not a true distance metric, as it is non-symmetric (i.e. $\text{HD}(X, Y) \neq \text{HD}(Y, X)$). To overcome this, one can use the symmetric HD:

$$\text{Symmetric HD}(X, Y) = \max(\text{HD}(X, Y), \text{HD}(Y, X))$$

For all further experiments and results, when reporting HD we are abusing the term and using the symmetric HD. A lower HD is indicative of a model that segments with borders closer to ground truth.

We use the implementation provided by Google DeepMind [\[45\]](#).

4.3.3 Mean absolute distance

The mean absolute difference (MAD) is another surface metric, defined as:

$$\text{MAD}(X, Y) = \text{mean}(|X - Y|)$$

Since we are taking the average of the absolute differences, this metric gives us a more global understanding of the average surface distance. In contrast, the HD measures the maximum distance between our surfaces, and is therefore more affected by poor segmentations with outlier points.

5 Experiments and results

5.1 Datasets

We use the Automatic Cardiac Diagnosis Challenge (ACDC) [7] and the Multi-Centre, Multi-Vendor and Multi-Disease Cardiac Segmentation Challenge (M&Ms) [13] datasets. These datasets are popular in the literature, and have been used to investigate a variety of supervised and unsupervised segmentation methods. They are both inherently sparse (across volumes) as they provide annotations at only end-diastolic and end-systolic phases. Both datasets provide pixel-level annotations for 3 foreground classes: the left ventricle, the myocardium and the right ventricle.

ACDC has a set of 100 training cases, each of which has two fully annotated volumes (one at end-diastole, one at end-systole). We train on all 200 volumes using 160 volumes for training and the remaining 40 volumes for validation. The test set is composed of 100 cases, each of which is again fully annotated at end-diastole and end-systole. After nnU-Net pre-processing transformations, all volume have 20 slices. The total number of available training slices is therefore $20 \times 160 = 3200$. Aside from the segmentation labels, patients within the ACDC dataset are classified into one of several cardiac conditions: dilated cardiomyopathy, hypertrophic cardiomyopathy, myocardial infarction with altered left ventricular ejection fraction, abnormal right ventricle and patients without cardiac disease.

Similarly, M&Ms has a set of 150 training cases, each of which has end-diastole and end-systole volumes annotated (300 volumes) [1]. We split the training data into 240 training cases and 60 validation cases. The test set is composed of 136 cases (again, each case has end-diastole and end-systole annotated) [2]. After nnU-Net pre-processing transformations, there are 14 slices per volume - the total number of available training slices is therefore $14 \times 240 = 3360$. The M&Ms dataset contains images from four different vendors (Siemens, Philips, General Electric and Canon) and six clinical centers. We assign the same patients to the train/validation/test splits as outlined in the original paper [13].

5.2 Sparse data

We investigate data sparsity by training networks on limited patients. For the ACDC dataset, we train networks on 8, 24, 32, 48, 80 and 160 patients. As a result of more patients within the M&Ms dataset, we additionally train on 192 and 240 patients. The results are shown in Table 5.1.

¹Although the original publication specifies that there are 175 training cases, 25 of these are unlabelled.

²The original publication specifies 160 testing patients - it is unclear why there are missing data.

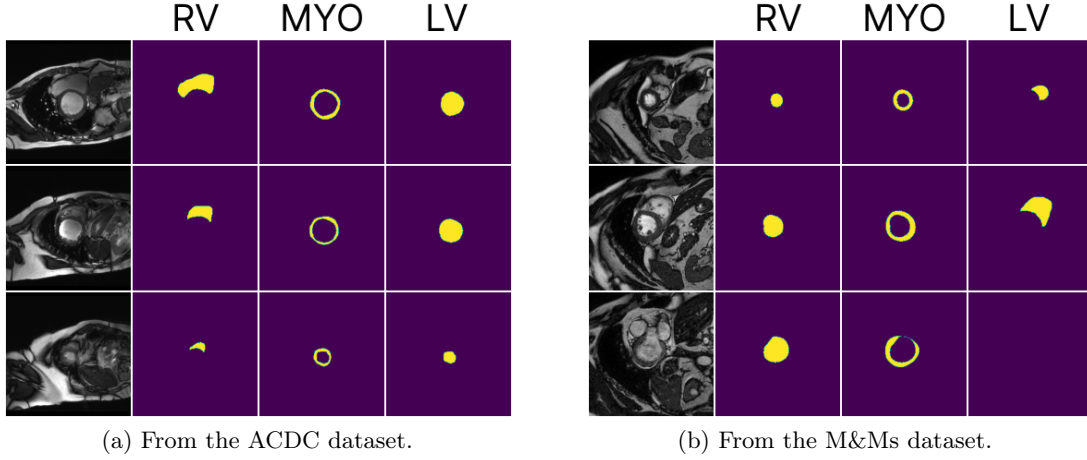


Figure 5.1: Example cMRI images, with slice annotations. From top to bottom are an apical slice, a mid-ventricular slice and a basal slice. As we move from the basal to the apical region the size of the cardiac structures decrease. RV = right ventricle, MYO = myocardium and LV = left ventricle.

Dataset	Network	Evaluation metric	Number of training volumes										
			1	8	16	24	32	48	80	120	160	192	240
ACDC	2D	Dice	0.39	0.62	0.71	0.71	0.74	0.85	0.89	0.90	0.91	-	-
		HD (mm)	101.37	36.03	25.95	22.15	18.56	14.61	7.20	5.35	5.06	-	-
		MAD (mm)	30.77	10.16	7.20	5.67	8.95	2.81	1.65	1.36	1.16	-	-
	3D	Dice	0.32	0.57	0.63	0.66	0.71	0.85	0.85	0.89	0.91	-	-
		HD (mm)	109.68	54.93	54.63	43.63	29.34	8.81	7.75	6.20	4.40	-	-
		MAD (mm)	37.82	18.03	16.70	13.61	8.95	2.17	1.94	1.56	1.16	-	-
M&Ms	2D	Dice	0.13	0.60	0.79	0.82	0.83	0.85	0.85	0.86	0.86	0.86	0.87
		HD (mm)	114.14	32.43	21.94	9.30	9.42	8.81	7.75	7.11	6.84	6.87	6.54
		MAD (mm)	43.74	8.90	4.81	2.39	2.38	2.17	1.94	1.75	1.74	1.74	1.74
	3D	Dice	0.19	0.54	0.78	0.82	0.82	0.85	0.85	0.86	0.86	0.86	0.87
		HD (mm)	109.55	37.41	18.00	9.10	8.86	6.98	6.44	6.20	5.89	5.80	6.02
		MAD (mm)	40.91	11.26	4.53	2.25	2.32	1.79	1.65	1.57	1.56	1.53	1.60

Table 5.1: Effect of training different nnU-Nets on sparse annotated volumes. Note that the ACDC dataset only has 160 volumes.

The best performing networks are those trained on all volumes - networks trained on ACDC achieve a maximum Dice score of 0.91. Those trained on M&Ms achieve a maximum of 0.87. Using more than 48 volumes gives a Dice score equal to 0.85 for all networks on both datasets (approximately 30% of the ACDC dataset, and 20% of M&Ms). Further decreasing the number of training volumes leads to decreased Dice scores and increased surface distances. However, we note that even when using very little data, networks trained on M&Ms are still able to perform very well (both 2D and 3D networks trained on 24 volumes - one tenth of total data - achieve a Dice score within 5% of networks trained on

the entire dataset). Qualitative difference between 3D networks trained on one tenth of available volumes is shown in Figure 5.2.

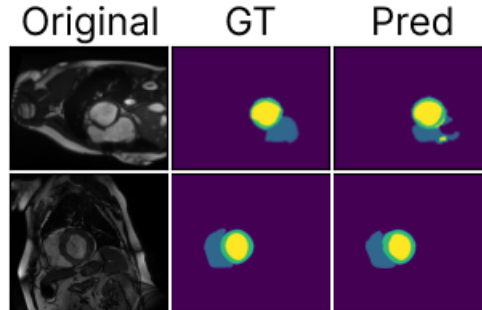


Figure 5.2: Qualitative difference between 3D nnU-Nets trained on one tenth of available volumes. The top row shows a slice from the ACDC dataset, predicted by a network trained on 16 volumes (with 20 slices per volume, for a total of 320 slices). The bottom row show a slice from the M&Ms dataset, predicted by a network trained on 24 volumes (with 14 slices per volume, for a total of 336 slices).

5.3 Sparse annotations

The results of training on particular cardiac regions is shown in Table 5.2. A qualitative example showing the difference in performance for networks trained on limited regions is shown in Figure 5.3. As expected, the best results are achieved when using all 3 cardiac regions (i.e. the most slices).

All networks trained on two regions achieve a Dice score greater than 0.82, except for the network trained on M&Ms data on apical and basal slices, which achieves a score of 0.78. For both datasets, networks trained with mid-ventricular slices (i.e. networks trained with apical and mid-ventricular or basal and mid-ventricular slices) achieve the highest scores. Training on a combination mid-ventricular and basal slices achieves results within 2% of using all available slices on the ACDC dataset and within 3% on the M&Ms dataset.

Networks trained on only a single region perform the worst. Of these, networks trained on only apical slices perform particularly poorly. Networks trained on only mid-ventricular slices perform the best, achieving Dice scores of 0.75 on the ACDC dataset and 0.65 on the M&Ms dataset. These two networks, however, have very high surface distances (a HD of 49.12mm on ACDC and 96.10mm on M&Ms), indicating spurious segmentations (an example is shown in Figure 5.4).

We then train on randomly sampled slices from the entire cardiac volume. The results are shown in Table 5.3. Again, we observe how best results are achieved using all available slices. To compare to training on particular cardiac regions, we note that one cardiac region corresponds to using one-third of available slices. For ACDC, training on a single cardiac region can be compared to training on 6 random slices; training on two cardiac regions can be compared to 13 slices. For M&Ms, a single cardiac region compares to 5 slices, and two regions to 10 slices.

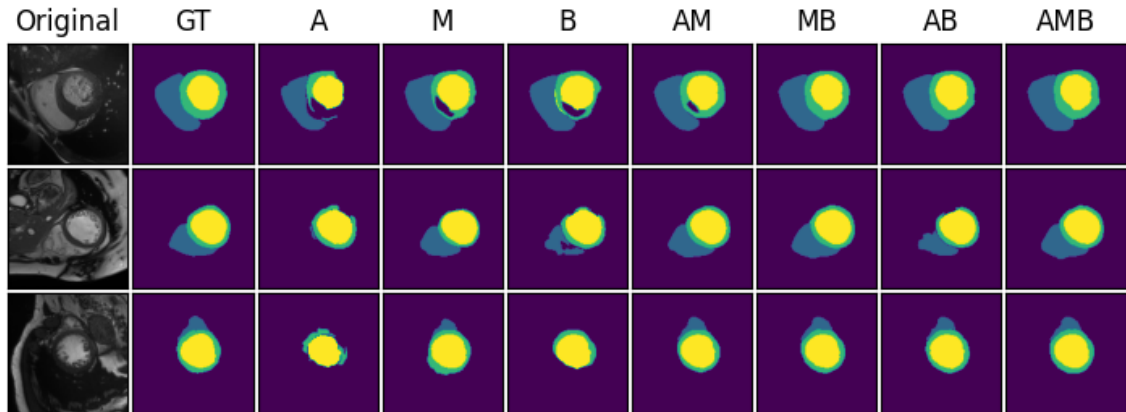


Figure 5.3: Qualitative results of segmentations produced by networks trained on different cardiac regions on the ACDC dataset. From top to bottom are slices from the basal region, the mid-ventricular region and the apical region respectively. Each slice is extracted from a different 3D segmentation volume. GT = ground truth. AMB means the network was trained on apical/middle-ventricular/basal slices (and permutations thereof).

Dataset	Metric	Cardiac regions trained on						
		A	M	B	A + M	M + B	A + B	A + M + B
ACDC	Dice	0.53	0.75	0.55	0.86	0.89	0.82	0.91
	HD (mm)	55.99	49.12	37.56	7.10	5.46	9.52	4.17
	MAD (mm)	21.43	18.78	12.38	1.76	1.58	2.66	1.08
M&Ms	Dice	0.04	0.65	0.65	0.82	0.84	0.78	0.87
	HD (mm)	51.75	96.10	29.16	10.00	7.53	10.50	5.52
	MAD (mm)	24.10	33.62	11.65	3.60	1.94	2.82	1.47

Table 5.2: Effect of training a **3D nnU-Net** with sparsely annotated cardiac regions (A=apical slices, M=middle slices, B=basal slices). The network is evaluated on all regions.

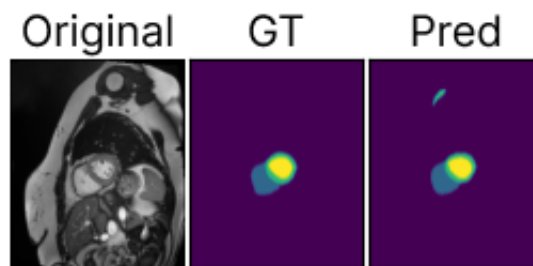


Figure 5.4: Resulting spurious segmentation for a 3D nnU-Net trained only on mid-ventricular slices on the ACDC dataset.

Networks trained on a random third of all slices out-perform those trained on any single cardiac region. Networks trained on a random sample of two thirds of slices achieve results comparable to training on the combination of mid-ventricular and basal slices.

We further observe a Dice score of greater than 0.8 when using 40% of ACDC slices and approximately 60% of M&Ms slices. Using more than 10 slices achieves a score greater than 0.85 for both datasets - this corresponds to using half of ACDC slices and approximately 70% of M&Ms slices.

		Number of slices used for training											
Dataset	Metric	1	2	4	5	6	7	8	10	13	14	16	20
ACDC	Dice	0.01	0.28	0.62	0.75	0.77	0.82	0.81	0.87	0.89	0.90	0.90	0.91
	HD (mm)	82.18	24.79	19.34	11.07	10.83	9.21	8.58	6.88	4.88	5.00	4.78	4.63
	MAD (mm)	33.09	8.34	6.18	3.46	2.95	2.80	2.39	1.71	1.29	1.30	1.22	1.20
M&Ms	Dice	0.01	0.28	0.68	0.72	0.79	0.79	0.83	0.85	0.85	0.86	-	-
	HD (mm)	76.06	28.2	14.10	12.48	11.11	8.82	7.58	6.33	6.54	5.64	-	-
	MAD (mm)	35.3	10.31	4.27	3.80	3.82	2.79	2.40	1.67	1.62	1.49	-	-

Table 5.3: Influence of training with randomly sampled and sparsely annotated slices from all three cardiac regions using 3D nnU-Net. Note that there are only 14 slices per volume (after pre-processing) for the M&Ms dataset.

5.4 Sparse data vs sparse annotations

To determine the relative importance of data compared to slices for performance, we run a series of experiments where we hold the total number of slices constant while varying the proportion of data volumes to slices. Table 5.4 shows the results for approximately 1400 slices annotated. Both datasets show the same trend of improved results when annotating more slices per volume. This is shown qualitatively in Figure 5.5.

To further investigate the importance this relationship between more data vs more slices we halve the number of volumes, training networks on a total of approximately 700 slices. The results are shown in Tables 5.5 and 5.6. Again, we observe that annotating more slices increases performance, even if a smaller number of volumes is used. When using the same number of volumes and slices, networks trained on ACDC seem to perform better than those trained on M&Ms. We also note that networks trained on ACDC seem to be more affected by changes in the number of available volumes (for example, we see large changes in Dice scores when using 9 and 20 slices and doubling the volumes).

We observe that generally doubling the number of annotated volumes lead to little improvement in evaluation metrics (with exception to networks trained on ACDC with 9 or 20 slices). In contrast, increasing the number of annotated slices can have a very large effect - we note an increase of 9% on ACDC when annotating 12 slices compared to 9 slices, and an increase of 8% on M&Ms when annotating 10 slices compared to 6 slices.

		Proportionality constant $V \times S = \sim 1400$				
Dataset	Metric	65 V, 20 S	100 V, 14 S	120 V, 12 S	160 V, 9 S	240 V, 6 S
ACDC	Dice	0.87 ± 0.02	0.88 ± 0.02	0.88 ± 0.03	0.82 ± 0.09	-
	HD (mm)	8.00 ± 1.47	8.52 ± 0.72	6.70 ± 0.5	8.27 ± 1.45	-
	MAD (mm)	1.98 ± 0.3	2.08 ± 0.18	1.60 ± 0.08	2.01 ± 0.33	-
M&Ms	Dice	-	0.86 ± 0.03	0.84 ± 0.02	0.83 ± 0.06	0.77 ± 0.08
	HD (mm)	-	6.10 ± 0.95	6.72 ± 1.25	6.93 ± 0.59	9.28 ± 0.78
	MAD (mm)	-	1.58 ± 0.16	1.86 ± 0.15	1.91 ± 0.22	2.75 ± 0.23

Table 5.4: Segmentation performance of a 3D nnU-Net when changing the number of training volumes (V) vs the number of slices (S) while keeping the total number of slices approximately 1400. Note that the M&Ms dataset has a total of 14 slices per volume, and that the ACDC dataset has a total of 160 volumes.

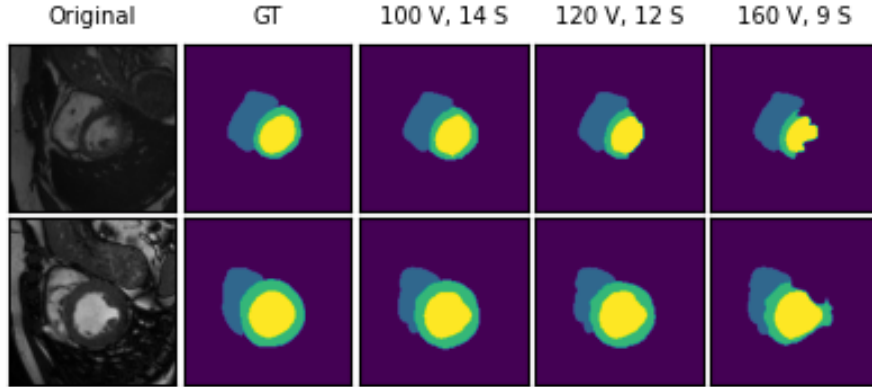


Figure 5.5: Qualitative results demonstrating improved performance as the network is trained on an increased number of annotated slices/decreased number of volumes. The first row is a random slice from a 3D segmentation on the ACDC dataset. The second row is the same on the M&Ms dataset. V=volumes, S=slices and represents the total number of volumes/slices each network was trained on respectively.

Slices	9		10		12		14		17		20	
Volumes	80	160	77	144	60	120	50	100	40	80	32	65
Dice	0.78	0.82	0.85	0.87	0.87	0.88	0.87	0.88	0.84	0.89	0.70	0.87
HD (mm)	9.61	8.27	8.50	7.67	9.21	6.70	8.83	8.52	11.35	6.15	32.4	8.00
MAD (mm)	2.41	2.01	2.16	1.81	2.18	1.60	2.35	2.08	3.59	1.51	9.79	1.98

Table 5.5: Influence of keeping slices constant while reducing number of training volumes. Trained on **ACDC** with a 3D nnU-Net network.

Slices	6		8		9		10		12		14	
Volumes	120	240	96	192	80	160	77	144	60	120	50	100
Dice	0.75	0.77	0.81	0.82	0.82	0.83	0.83	0.84	0.85	0.84	0.84	0.86
HD (mm)	12.11	9.28	7.77	8.17	7.79	6.93	6.92	7.07	6.30	6.72	7.14	6.10
MAD (mm)	3.76	2.75	2.15	2.34	2.22	1.91	1.83	1.93	1.67	1.86	1.79	1.58

Table 5.6: Influence of keeping slices constant while reducing number of training volumes. Trained on M&Ms with a 3D nnU-Net network.

5.5 Segment Anything Model (SAM)

To determine the best prompt setup for fine-tuning SAM, we ran a series of inference experiments on a baseline model. We test baseline performance using permutations of positively sampled points, negatively sampled points and presence/absence of bounding boxes. As outlined in Chapter 4, not using any input prompts leads to poor results. Figure 5.6 shows baseline SAM predictions with a variety of prompts.

In general, the less information we use for prompting the more comparable inference is to a U-Net. We would prefer to only use positive sample points. However, we noticed that SAM would frequently over-segment the myocardium. As shown in Figure 5.1, the myocardium largely envelops the left ventricle, and the differentiation between these structures is often unclear. SAM frequently segments both the left ventricle and myocardium together, as a single structure. To solve this issue, we experiment with using negative sample points (i.e. points that are explicitly not part of the class to segment). The benefits of this can be seen in Figure 5.7. Although we could sample new negative points, we prefer to re-use previously sampled positive points. For example, if we have sampled some positive points for the left ventricle, these points can be re-used as negative samples when running inference on the myocardium or right ventricle. Re-using positive samples as negative samples for other classes is more efficient and translates into less effort for an end-user.

The results for inference averaged over 5 runs on ACDC and M&Ms are shown in Tables 5.7 and 5.8 respectively. We note consistent and similar results for both datasets. Worst results use the fewest positive samples, without any negative samples and without bounding boxes. Better results are achieved by using more positive samples, using negative samples and/or using bounding boxes. We also observe that not using bounding boxes leads to significantly increased surface metrics, and the uncertainty of those metrics.

On networks inferred on the ACDC dataset, the best Dice score of 0.71 is achieved without bounding boxes, using five positive sample points and two negative sample points. Inferring with bounding boxes gives a very similar result, achieving a Dice score of 0.70. Networks inferred on the M&Ms dataset achieve a best Dice score of 0.70 using bounding boxes, with three positive samples and two negative samples. On both datasets, the best surface metrics are achieved by prompting with bounding boxes, using two positive sample points and one negative sample point.

We choose to fine-tune the ‘cheapest’ prompts (in terms of cost per prompt) - those with two positive points and one negative sample. Despite being the cheapest to annotate, these models achieve results close to models that are prompted with more positive points.

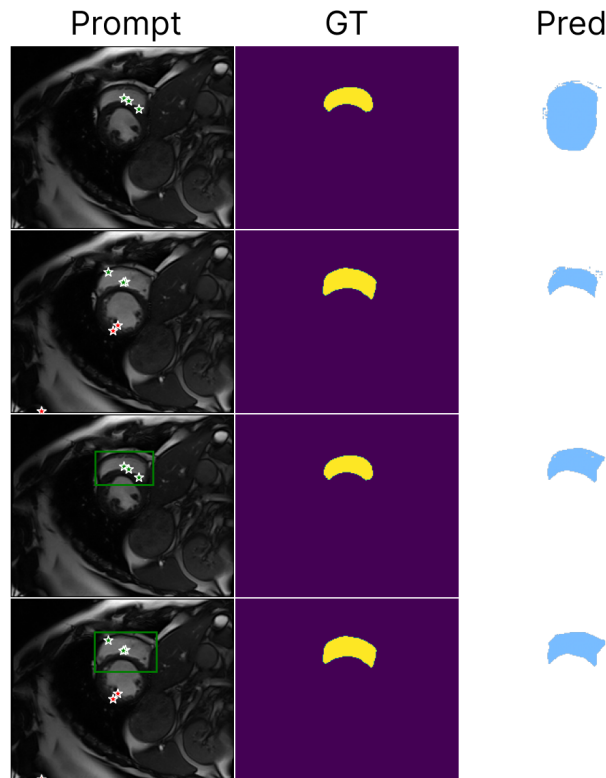


Figure 5.6: Segmentation results on the right ventricle, generated by baseline SAM on the M&Ms dataset. From top to bottom the model is prompted with only positive points, positive and negative points, positive points with a bounding box and finally positive points, negative points and a bounding box. Positive and negative samples are shown as green and red stars respectively.

We fine-tune models both with and without bounding boxes to compare performance.

Models were fine-tuned for 100 epochs. We use the Adam optimiser ($\beta_1 = 0.9, \beta_2 = 0.999$). An initial learning rate of $1e-4$ was empirically determined. We reduce the learning rate on epoch loss plateau with a patience of 5 by a factor of 0.1. A batch size of 1 was used due to GPU memory limitations. The same training and validation transformations as nnU-Net, including spatial transforms, adding Gaussian noise, mirroring, etc. were applied. Dice cross entropy loss was used, with equal weighting given to both loss terms. MONAI [46] and Pytorch [47] were used to build the fine-tuning pipeline.

Results (averaged over 5 runs) of fine-tuning models with bounding boxes with sparse data are shown in Table 5.9. Results for fine-tuning without bounding boxes are shown in Tables 5.10 and 5.11. Models are evaluated on all available testing data. For both datasets we observe that fine-tuning with bounding boxes improves performance, with best performance achieved by training using all available training data. However, fine-tuning without bounding boxes yields models that perform worse than baseline. Without using bounding boxes, models seem unable to learn (and actively become worse).

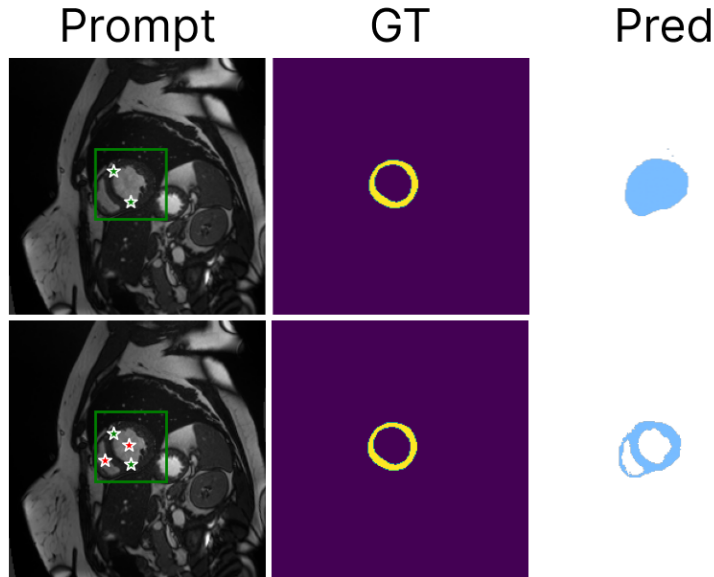


Figure 5.7: Effect of using negative samples when running inference on the myocardium. The top row shows sampling only using a bounding box and two positive myocardium samples (shown as green stars). We note the resulting over-segmentation. The bottom row shows the same, but with a negative sample from each of the ventricles (red stars); while not perfect, the segmentation is much more accurate.

On the ACDC dataset, we observe a 15% increase in Dice score and a 6.43mm reduction in HD between baseline inference without any fine-tuning and using a model fine-tuned with bounding boxes using all available data. On the M&Ms dataset the increase is more modest, with a 7% improvement in Dice and a 5.74mm improvement in HD. For both datasets we observe improvement of MAD scores of about 1.5mm. Using only 8 annotated volumes leads to a 7% increase in Dice on both datasets. Qualitative results comparing baseline to fine-tuned models are shown in Figures [5.8](#) and [5.9](#). Further qualitative results of models fine-tuned without bounding boxes are shown in Figure [5.10](#).

Bounding boxes	Pos. samples	Neg. samples	Dice	HD (mm)	MAD(mm)
N	2	0	0.48	40.21 ± 14.98	16.39 ± 6.52
		1	0.67	13.94 ± 8.58	7.43 ± 2.69
		2	0.66	14.65 ± 8.68	7.20 ± 2.70
N	3	0	0.50	36.57 ± 14.55	15.08 ± 5.86
		1	0.69	13.42 ± 7.57	7.41 ± 2.40
		2	0.68	13.80 ± 8.30	7.45 ± 2.90
N	5	0	0.58	27.40 ± 13.64	12.30 ± 5.00
		1	0.69	13.37 ± 7.37	7.67 ± 2.72
		2	0.71	13.08 ± 8.28	7.83 ± 3.19
Y	2	0	0.64	14.76	7.35
		1	0.69	12.21	6.99
		2	0.70	12.22	7.31
Y	3	0	0.64	14.78	7.37
		1	0.69	12.66	7.10
		2	0.69	12.86 ± 5.11	7.35
Y	5	0	0.64	14.61	7.34
		1	0.68	13.01	7.17
		2	0.70	13.44 ± 5.64	7.72 ± 2.23

Table 5.7: Baseline SAM inference results on the **ACDC** dataset. Positive and negative sample counts are per each segmentation class. All Dice standard deviations are less than 0.15. Unless shown, HD standard deviations are less than 5mm and MAD standard deviations are less than 2mm.

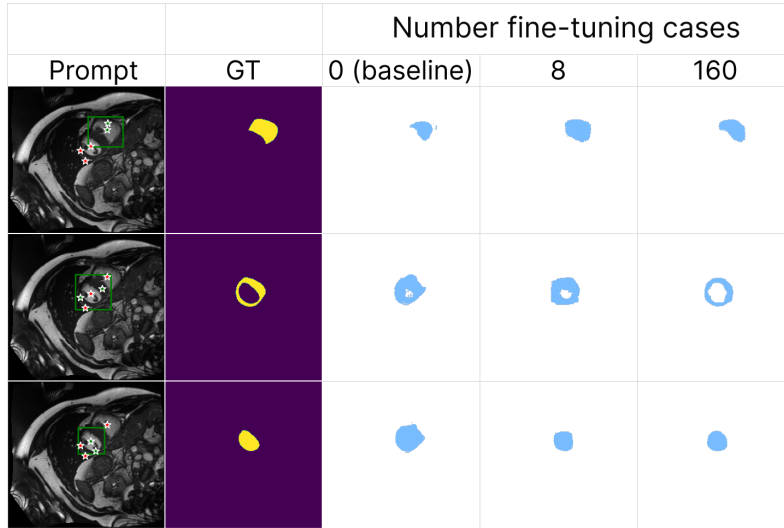


Figure 5.8: Qualitative results showing the difference in outputs of SAM models fine-tuned on the **ACDC** dataset. Models were fine-tuned with bounding boxes, using 2 positive points and 1 negative point per class.

Bounding boxes	Pos. samples	Neg. samples	Dice	HD (mm)	MAD(mm)
N	2	0	0.46	40.79 ± 14.66	16.69 ± 6.57
		1	0.65	14.44 ± 8.59	7.46 ± 2.92
		2	0.64	15.38 ± 9.28	7.35 ± 3.11
N	3	0	0.48	37.04 ± 13.50	15.15 ± 5.62
		1	0.66	14.21 ± 8.44	7.71 ± 3.11
		2	0.67	14.31 ± 9.00	7.79 ± 3.65
N	5	0	0.57	27.09 ± 12.02	11.69 ± 4.25
		1	0.68	13.69 ± 7.74	7.56 ± 2.79
		2	0.68	13.77 ± 8.24	7.91 ± 3.36
Y	2	0	0.66	12.86	6.67
		1	0.69	11.67	6.57
		2	0.69	11.96	6.92
Y	3	0	0.66	12.86	6.66
		1	0.69	12.02	6.63
		2	0.70	12.24 ± 5.40	7.04
Y	5	0	0.66	12.75	6.64
		1	0.68	12.23	6.79
		2	0.69	12.77 ± 5.77	7.30 ± 2.15

Table 5.8: Baseline SAM inference results on the **M&Ms** dataset. Positive and negative sample counts are per each segmentation class. All Dice standard deviations are less than 0.15. Unless shown, HD standard deviations are less than 5mm and MAD standard deviations are less than 2mm.

		Number of fine-tuning cases								
Dataset	Metric	0 (baseline)	8	24	32	48	80	160	192	240
ACDC	Dice	0.69	0.76	0.81	0.82	0.82	0.83	0.84	-	-
	HD (mm)	12.21	9.92	7.42	7.06	7.65	6.33	5.78	-	-
	MAD (mm)	6.99	5.44	5.41	5.41	5.41	5.40	5.40	-	-
M&Ms	Dice	0.69	0.76	0.74	0.74	0.75	0.76	0.76	0.76	0.76
	HD (mm)	11.76	8.12	7.30	7.21	7.07	7.82	6.37	6.61	6.05
	MAD (mm)	6.57	5.14	5.13	5.13	5.13	5.16	5.13	5.13	5.12

Table 5.9: Inference results for SAM models fine-tuned with limited training data. The models were prompted **with bounding boxes**, two positive sample points and one negative sample per class. Dice standard deviations, HD standard deviations and MAD standard deviations are less than 0.1, 3mm and 1.2mm for all models respectively. Note that the ACDC dataset only has 160 volumes.

Metric	Number of fine-tuning cases						
	0 (baseline)	8	24	32	48	80	160
Dice	0.67	0.61	0.57	0.57	0.58	0.59	0.56
HD (mm)	13.94 ± 8.58	17.11 ± 6.10	20.37 ± 7.64	23.32 ± 10.15	19.05 ± 6.93	19.56 ± 8.31	19.30 ± 6.03
MAD (mm)	7.43 ± 2.69	7.55	8.62 ± 2.36	10.40 ± 4.33	7.67	8.89 ± 3.18	8.46

Table 5.10: Inference results of a SAM model, fine-tuned on the **ACDC** dataset. The models were prompted **without bounding boxes**, two positive sample points and one negative sample per class. Dice standard deviations are less than 0.1 for all models. Unless shown, MAD standard deviations are less than 2mm.

Metric	Number of fine-tuning cases								
	0 (baseline)	8	24	32	48	80	160	192	240
Dice	0.65	0.54	0.62	0.59	0.57	0.58	0.58	0.59	0.60
HD (mm)	14.44 ± 8.59	23.91 ± 8.49	15.71 ± 5.69	17.15 ± 5.29	18.05 ± 5.32	17.75 ± 5.35	17.89 ± 5.70	17.33 ± 5.57	16.71 ± 6.04
MAD (mm)	7.46 ± 2.92	11.14 ± 3.81	6.60	7.76	7.77	7.77	7.51	7.41	7.23

Table 5.11: Inference results of a SAM model, fine-tuned on the **M&Ms** dataset. The models were prompted **without bounding boxes**, two positive sample points and one negative sample per class. Dice standard deviations are less than 0.1 for all models. Unless shown, MAD standard deviations are less than 1.8mm for all models respectively.

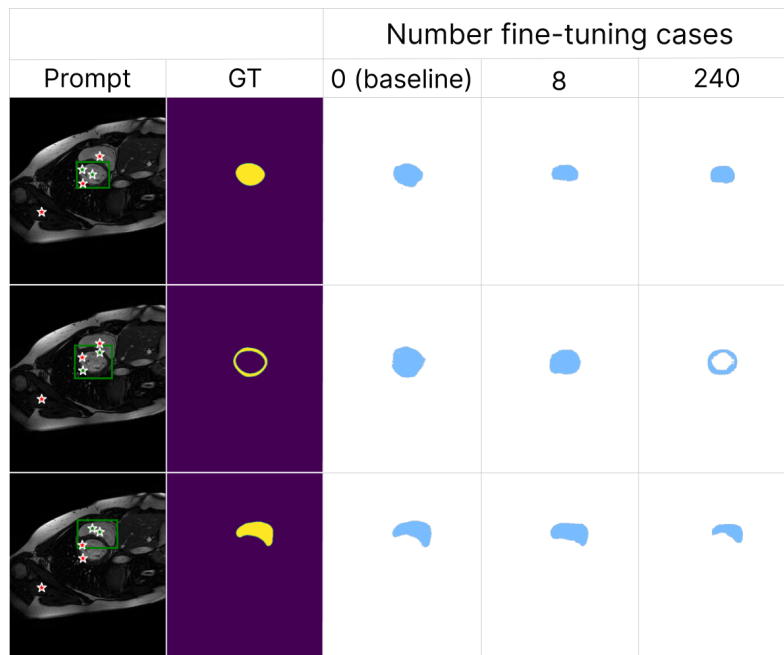


Figure 5.9: Qualitative results showing the difference in outputs of SAM models fine-tuned on the **M&Ms** dataset. Models were fine-tuned with bounding boxes, using 2 positive points and 1 negative point per class.

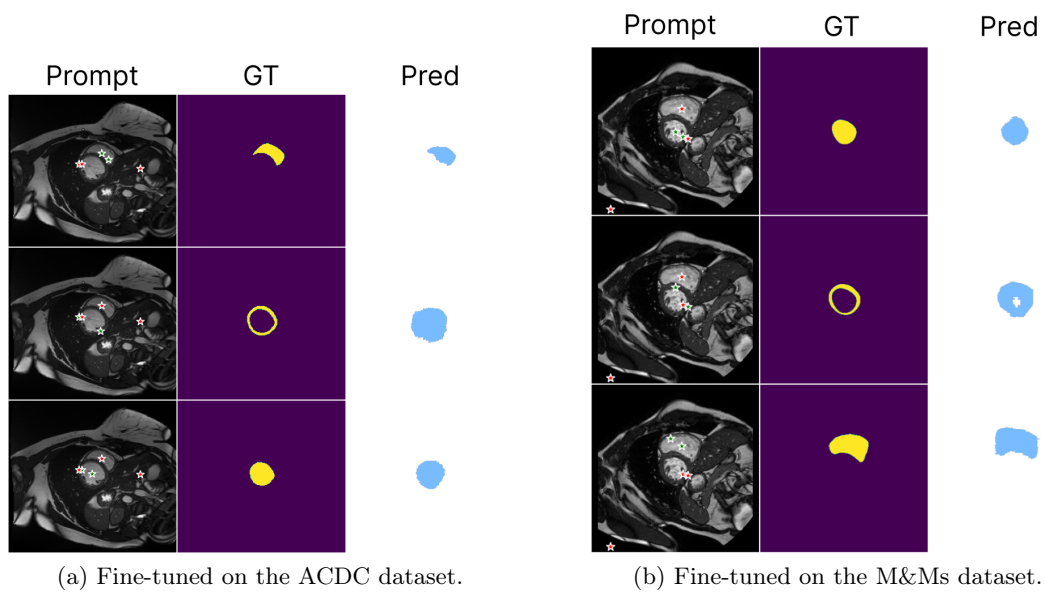


Figure 5.10: Qualitative results showing the difference in outputs of fine-tuned SAM models prompted **without bounding boxes**, using 2 positive points and 1 negative point per class. Models were trained with all available data.

6 Discussion

6.1 Sparse data

As shown in Table [5.1](#) using more training data allows for better performance. Surprisingly, we are able to severely limit the amount of training data before seeing a drop in performance. General deep-learning advice tells us that more data are always better - in this case, we see that while this is true, the extent of the value this additional data bring are limited.

For example, when halving the total number of training volumes, the drop in Dice score for 2D networks trained on the ACDC dataset is only 2%. Similarly, the drop for 2D networks trained on M&Ms is only 1%. The cost of time taken to annotate significantly more training volumes is expensive, yet comes at little benefit.

The 3D networks trained on ACDC are more affected by data reduction - we observe a reduction in Dice score of 6% when training with half the total number of training volumes. In contrast, those trained on M&Ms seem to be more robust, with the Dice score dropping by only 2%. In general, we note that the difference in performance between 2D and 3D networks is greater (more affected) for networks trained on ACDC compared to those trained on M&Ms. As we reduce the number of training volumes, the gap in Dice score between 2D and 3D networks trained on the ACDC dataset increases, while the gap between networks trained on the M&Ms dataset remains fairly consistent.

Annotating upwards of 48 volumes (regardless of the model dimensionality or dataset) yields networks that achieve Dice scores greater than 0.85. 48 volumes is approximately 30% of the ACDC dataset and approximately 20% of the M&Ms dataset. A score this high could be suitable for some clinical tasks. Again, thought should be given to the cost of annotating significantly more data relative to the improvement in performance.

Networks trained on M&Ms, a dataset with much more data variability (due to having multiple domains within the dataset), are particularly robust to training with limited data. Using only 10% of all volumes still yield networks that achieves Dice scores of 0.82 (within 5% of training with all available data). In contrast, when training with 10% of data, networks trained on ACDC achieve scores of 0.71 (a 20% drop in performance) and 0.63 (a 28% drop in performance) for 2D and 3D networks respectively. The exception to this is training with only a single volume - although networks trained on both datasets perform poorly, those trained on ACDC outperform those trained on M&Ms. We believe that having more variable data within the M&Ms dataset makes it more difficult to achieve a high Dice score (even when many volumes are available) while simultaneously allowing for good generalisation performance even when the number of training volumes is restricted.

6.2 Sparse annotations

As shown in Tables 5.2 and 5.3, using more slices for training yields the best performing networks. Again, the amount of slices we can reduce before seeing significant impact is surprising. For networks trained on the ACDC dataset, using half the available slices gives a Dice score within 4% of using all available slices. For those trained on M&Ms, we see a score within 7%.

Performance starts to deteriorate at around 8 slices for networks trained on ACDC and around 6 slices for those trained on M&Ms (i.e. around 40% of slices for both datasets). Training on upwards of 13 slices on the ACDC dataset (65% of available slices) and upwards of 10 slices on the M&Ms dataset (70% of available slices) both yield excellent networks that are comparable to training with all data (achieving Dice scores within 2% of networks trained on all slices).

It is interesting to observe that while networks trained on the M&Ms dataset were more robust to training with sparse data, it seems that they are more sensitive to training with sparse annotations. That is, the networks trained on the M&Ms dataset have larger drops in Dice score for the same proportion of annotated slices compared to those trained on the ACDC dataset.

Our investigations into training on slices from restricted cardiac regions show that networks trained with mid-ventricular slices (either alone or in combination with other slices) achieve the highest Dice scores and lowest surface distances. Within mid-ventricular slices, the cardiac structures are particularly large and well-delineated. Additionally, the slices at the top and bottom of the mid-ventricular region may strongly resemble basal and apical slices respectively. We hypothesise that having large mid-ventricular annotations present rich, information-dense structures for networks to learn from. Additionally, the networks are able to learn some basal and apical features from the top and bottom mid-ventricular slices. In contrast, apical and basal slices often show smaller, less information-dense structures, and therefore yield networks that do not perform as well. This is particularly true of apical slices.

When comparing training on a single region to training on a third of randomly sampled slices, we observe that using random samples yields higher-performing networks. For networks trained on the ACDC dataset, training on a random third of slices (approximately 6 slices) yields a network that achieves a Dice score 2% higher than the best network trained on a single cardiac region (with Dice scores of 0.77 and 0.75 respectively). On the M&Ms dataset, the network trained with a random third of slices (approximately 5 slices) achieves a Dice score 7% higher than the best network trained on a single cardiac region (with Dice scores of 0.72 and 0.65 respectively). Despite the somewhat modest improvements in Dice scores, we observe very strong improvements in surface distance metrics. On the ACDC dataset, we observe a 38.28mm improvement in MD and a 15.83mm improvement in MAD comparing between the network trained on mid-ventricular slices and the network trained with a random third of slices. On the M&Ms dataset, we observe even stronger improvements of 83.62mm in HD and 29.82mm in MAD. Although the Dice scores are similar, we observe that training using random slices yields much more refined segmentations, with far fewer spurious structures.

Comparing training on two cardiac regions to training on two thirds of random slices,

we observe less significant performance changes. On the ACDC dataset, training on two thirds of random slices (approximately 13 slices) yields a network that is similar to the network produced by training on mid-ventricular and basal slices, with equal Dice scores of 0.89, but slightly improved surface metrics (we observe a 0.58mm improvement in HD and a 0.29mm improvement in MAD). On the M&Ms dataset, the network trained on two thirds of random slices (approximately 10 slices) is again similar to the network trained on mid-ventricular and basal slices, achieving a 1% higher Dice score, a 0.05mm increase in HD and a 0.46mm increase in MAD. Overall, there is little difference between training on two thirds of random slices compared to training on slices only from the mid-ventricular and basal regions. That being said, the cost of annotating only the mid-ventricular and basal slices might be lower, since the cardiac structures are larger and better differentiated, and may therefore be easier to segment.

Overall, we observe that random sampling tends to out-perform networks trained on particular cardiac regions. Random sampling allow networks to learn from slices throughout the entire cardiac volume; it makes sense that having more diversity in slices allow networks to learn more and achieve better results.

6.3 Sparse data vs sparse annotations

Tables 5.4, 5.5 and 5.6 show the effects of training with different proportions of volumes to slices. We observe that annotating more slices usually has a more significant effect on performance improvement compared to annotating more slices. Intuitively, this makes sense - annotating more slices allow networks to learn from more novel, spatially rich slices. In contrast, limiting the number of slices forces networks to learn from sparse representations, which fail to generalise well.

Table 5.4 shows that, when limiting the total number of slices to 1400, the best Dice scores on both datasets are achieved training with 100 volumes and 14 slices. This is the maximum number of slices on the M&Ms dataset, and so is consistent with our previous results showing that using the most slices gives the best performing networks (see Section 5.3). It is interesting to note, however, that on the ACDC dataset a network trained with all available slices (i.e. 20 slices) is not the best performing network. Again, from our previous experiments on sparse annotations, we know that using upwards of 13 slices on the ACDC dataset yields comparable networks to training on all available slices - indeed, networks trained on all 20 slices out-perform those trained on 14 slices by only 1%. Further, from our experiments on reduced data (see Section 5.2) we know that the networks trained on the ACDC dataset are more susceptible to using fewer training volumes. We therefore hypothesise that the worse performance obtained by the network trained with all 20 slices (and 65 volumes) is a result of the extra slices not yielding significant benefit, while at the same time the reduced number of volumes causing significant deterioration.

This is further exemplified in Table 5.5. Even when training on ACDC with all available slices, reducing the number of training volumes has a drastic effect - we observe a 17% drop in Dice score for networks trained with 32 compared to 65 volumes. Contrast this to the networks trained on the M&Ms dataset. From our experiments on limiting data, we know that these networks are more robust to reduced data. As shown in Table 5.6 the

best performing network is trained on the most available slices; even when we then reduce the number of slices by half, we see only a 2% drop in Dice score.

Especially when using a reduced number of slices, we observe that the performance improvements gained by increasing the number of slice annotations outweighs the improvements gained by annotating more volumes. Table 5.5 shows how, when training on the ACDC dataset with less than 12 slices, the performance improvement in Dice score obtained by doubling the number of training volumes is at most 4%. In contrast, the improvement gained from annotating 12 slices compared to 9 slices is a maximum of 9% (i.e. more than twice the improvement gained by annotating 3 more slices than by doubling the number of annotated volumes). Similarly, as shown in Table 5.6, on the M&Ms dataset the maximum Dice improvement gained by doubling the number of volumes is 2%. However, if we increase the number of slice annotations from 6 to 10 we observe a maximum Dice increase of 8% (a four-fold improvement increasing the number of annotated slices compared to doubling the number of annotated volumes).

6.4 Segment Anything Model (SAM)

6.4.1 Baseline inference performance

Tables 5.7 and 5.8 show that baseline SAM has limited performance, achieving Dice scores ranging from approximately 0.5 to approximately 0.7 on both datasets. This is inconsistent with previous work done by He et al. 41, who achieve baseline results (using a single positive sample point or bounding boxes with a margin of 20 pixels) ranging between approximately 0.2 and 0.4 on comparable datasets. It is, however, consistent with the values reported by Ma et al. 42.

We observe how using negative samples markedly improves performance. This is especially true when using fewer positive sample points without bounding boxes. When comparing networks trained with and without negative sample points (without bounding boxes), we see a maximum increase in Dice score of 19% on both datasets. This is inconsistent with the work of Huang et al. 40, where there is little difference in prompting with or without negative samples on their reported cMRI data.

The difference between using one or two negative samples is less pronounced. Often, there is no change in performance between using only a single point or two points. Occasionally the Dice score will change by a percentage.

Models inferred with two positive samples and negative samples achieve results close to models that use more positive samples. For example, the maximum Dice difference between inferring without bounding boxes using two positive and five positive samples is only 4% on the ACDC dataset. With bounding boxes, the difference is only 1%. Similarly, on the M&Ms dataset, the differences are 3% and 1%.

We note how using bounding boxes improves performance, especially when not using any negative sample points. For example, we see an increase in Dice score of 16% on networks trained on ACDC with two positive sample points with bounding boxes compared to those trained without. Similarly, we see an increase of 20% on those networks trained on M&Ms. As we increase the number of positive samples, the effect of bounding boxes becomes less pronounced. When using five positive sample points, the percentage increase without using

any negative samples drops to 6% for networks trained on the ACDC dataset and 9% for those trained on the M&Ms dataset. This is consistent with previous works of [39]–[42], who have all similarly shown improvements using bounding box prompting.

Although the Dice scores with and without bounding boxes are similar, we observe strong differences in surface metrics. When not using bounding boxes, the surface metrics are much higher, with much greater variability. This indicates spurious segmentations that are far from ground truth.

Using additional positive sample points seems to only have an effect when using neither bounding boxes nor negative sample points. For networks trained on the ACDC dataset, we note how using three additional positive samples increases the Dice score by 10%. For those trained on the M&Ms dataset, we observe a similar increase of 9%. Similar or greater improvements are achieved by Huang et al. [40] when increasing from a single positive sample to five positive samples.

We hypothesise that both negative sample points and bounding boxes provide rich spatial information that is lacking when only a few positive sample points are used. When either of these prompts are provided they have dramatic effects on improving performance. When both are provided we see little improvement compared to using either option exclusively. Note, however, that drawing bounding boxes requires more time and effort of the end-user. In contrast, as outlined in Chapter 4, when we use negative samples we actively re-use previously sampled positive points. In this sense, using negative samples comes at no additional cost, and is therefore the more preferred option for baseline inference.

6.4.2 Fine-tuning performance

As shown in Table 5.9, fine-tuning networks on the ACDC dataset with bounding boxes leads to improved models. Fine-tuning with all available training data leads to a 15% improvement in Dice score. Similarly to the experiments training nnU-Net with reduced data, we note that while best results are achieved using the most available data, very good fine-tuning results can still be achieved using a reduced dataset. In this case, we note that fine-tuning using 24 volumes, or 15% of available data, produces a network that achieves a Dice score of 0.81 (a 12% increase in Dice score compared to baseline performance). This is also shown qualitatively in Figure 5.8, where we observe improved segmentation masks with more available fine-tuning data. These improvements are similar to those achieved by MedSAM over baseline SAM [42].

Compared to previous experiments with 2D nnU-Net, we observe that training with all available volumes (with sparse prompt inputs) gives results similar to a nnU-Net trained from scratch on 48 volumes. In general, models fine-tuned with datasets of less than 32 volumes tend to out-perform the equivalent nnU-Net models trained from scratch. However, once more volumes are available, the nnU-Net models perform better.

When analysing the results for networks trained with bounding boxes on the M&Ms dataset (again shown in Table 5.9), we note a more modest improvement in performance. Here, the fine-tuned model trained with all available data achieves a Dice score of 0.76, a 7% improvement over the baseline model. Additionally, it seems that performance improvement saturates very quickly, with little differences between training with progressively more data. However, despite the quantitative metrics not improving, we do observe qualitative

improvements with more training data, especially when segmenting the myocardium (as shown in Figure 5.9). From our earlier experiments, we know that M&Ms is a more challenging dataset to perform well on. That being said, the SAM model fine-tuned on all 240 volumes fails to reach the performance of a nnU-Net trained on only 16 volumes (the models achieve Dice scores of 0.76 and 0.82 respectively).

In general, the structures within the M&Ms dataset are much smaller compared to those within the ACDC dataset. One contributing factor towards poor performance may be the small size of foreground classes relative to the background. It is known that SAM struggles with smaller structures. Nevertheless, more work remains to be done investigating the lack of improvement on this dataset.

We observe that fine-tuning without bounding boxes leads to significantly worse performance on both datasets. From our earlier inference results, we know that not using bounding boxes tends to give very large surface distances. It is possible that the produced segmentations are too large, and the model is not able to learn well-differentiated borders. Qualitatively, as shown in Figure 5.10, we see results without sharp boundaries and difficulty segmenting structures with holes (i.e. the myocardium). It is also possible that training without bounding boxes requires a more specialised training pipeline.

Finally we remark that if image encodings are pre-computed, foundation model fine-tuning is cheap and efficient. The results of fine-tuning on the ACDC dataset are indicative that these foundation models could offer competitive performance compared to models trained with sparse data inputs. Determining exactly when/if foundation models can be improved or when they will be unable to adapt to a novel downstream task is a challenging question that should be explored in further work.

7 Conclusion

This work has explored the limits of data sparsity when training state-of-the-art segmentation networks for left ventricle, right ventricle and myocardial short-axis view cMRI segmentation. We have experimented on nnU-Net models trained from scratch, as well as baseline and fine-tuned SAM foundation models. We have enforced data sparsity (limiting the overall number of training volumes), annotation sparsity (limiting the number of available slices to train with, and the cardiac regions those slices are from) as well as sparsity in the annotations themselves (by using SAM prompted with a variety of differing sparse inputs).

When studying data sparsity, we have consistently shown that good segmentation results can be achieved using a significantly reduced dataset. 2D nnU-Net models trained on half of the available ACDC and M&Ms volumes were able to achieve Dice scores within 2% and 1% of training with the entire dataset respectively. 3D models were able to achieve scores within 6% and 1% on the ACDC and M&Ms datasets respectively. Using more than 48 volumes, regardless of the model dimensionality or dataset, was enough to achieve a Dice score of 0.85. This corresponds to 30% of the ACDC dataset and 20% of the M&Ms dataset. A score this high could be suitable for many clinical applications, and thought should be given to the cost of annotating more volumes relative to the value those volumes might provide.

When studying annotation sparsity, we again observed results of networks trained on limited slices can approach those of networks trained with all available slices. We showed that using upwards of 65% of slices on the ACDC dataset, and upwards of 50% of slices on the M&Ms dataset, provided enough training data to allow models to obtain Dice scores within 2% of models trained with all available slices.

In our investigations into training on particular cardiac regions, we determined that training with mid-ventricular slices yielded the most performant networks. Training with apical slices yielded the worst networks - this is due to the differences in ventricular size compared to the mid-ventricular and basal slices. Additionally, we showed that networks trained on randomly sampled slices out-performed those trained on any constrained cardiac regions.

Overall, when studying the relative importance of annotating more slices compared to more volumes, we have showed that annotating more slices has a greater impact on performance. Training using all available training volumes, with limited slices, yield poor networks. In contrast, training using all available slices, even with limited volumes, yield performant networks. We also note that the cost of annotating additional volumes is high, but provides little additional benefit. In comparison, the cost of annotating additional slices is low, but can provide significant benefit. We therefore recommend annotating more slices per volume, rather than more volumes with fewer slices.

When evaluating the state-of-the-art SAM foundation model, we found that it provides

limited out-of-the-box segmentation results. We can improve the results by prompting the model. Prompting can be done with combinations of bounding boxes, positive sample points and/or negative sample points. Our analyses show that while each of these individually can have a significant impact, there is little difference gained by combining them. Using bounding boxes is potentially the most expensive option, but does not yield significantly better results on baseline inference. Re-using positive samples of one class as negative samples for other classes comes at no additional cost, but yields results very similar to using bounding boxes. For baseline segmentation, we therefore recommend using more positively sampled points that are re-used as negative samples.

We have further shown that foundation models can, given the appropriate setup, be fine-tuned for cMRI segmentation. By using 8 fully labelled samples we were able to improve baseline results by 7% on both datasets. On the ACDC dataset, we were able to improve baseline performance by 15%, achieving a final Dice score of 0.85. It is important, however, to ensure that the model is correctly prompted. We observed that fine-tuning without bounding boxes led to worse results than baseline, despite baseline inference giving comparable performance between models prompted with and without bounding boxes. Finally, we note that the cost of fine-tuning is relatively small, but can yield significant improvements. When pre-computing image encodings, fine-tuning is fast and efficient.

There are several opportunities to expand this work. First, sparsity on other datasets and with other modalities could be investigated. Second, other pre-trained networks could be compared against - for example, using networks that have been pre-trained on large amounts of unlabelled data using contrastive learning, or using non-foundation models that have been pre-trained on natural images. Third, a deeper investigation into the needed prompts for effectively fine-tuning foundation models could be performed. Fourth, more work could be done into developing foundation segmentation models that are able to deal with poorly differentiated, small-scale structures (often found in medical images). Finally, further work could be done into determining, a priori, how much data one needs to train a satisfactory network.

Acknowledgments

First and foremost, thank you to my family. Without their continuous love and support I would not have been able to embark on or finish this master's.

Thank you to Dr. Di Folco, for his continuous guidance, support and help. I have not always been an easy student, but Dr. Di Folco has fully supported me throughout our work together.

Thanks to Prof. Schnabel for her reassurances and assistance, as well as her proof readings and suggestions.

Thank you to the Mathigon team for the continuous support and encouragement. David, Phil, Gabe, Patrick, Kaira, Dymo, Sarah, Eda and Bec - thank you all for our weekly meetings and exciting feature releases. A special thanks to Philipp for being so understanding and accommodating, and for being a constant inspiration.

Finally, thank you to my friends and other loved ones for providing me with much-needed laughter, joy and upliftment. Jess, Jake, Miro, Sasha, Kartikay, David, Jakob, Adi and Ali, thank you all so much.

Bibliography

- [1] G. A. Roth, G. A. Mensah and V. Fuster, *The global burden of cardiovascular diseases and risks: A compass for global action*, 2020.
- [2] World Health Organization *et al.*, ‘Global health estimates: Disease burden by cause, age, sex, by country and by region, 2000-2019’, *Geneva. World Health Organization. Pobrano*, vol. 15, p. 2022, 2020.
- [3] G. A. Roth, G. A. Mensah, C. O. Johnson *et al.*, ‘Global burden of cardiovascular diseases and risk factors, 1990–2019: Update from the gbd 2019 study’, *Journal of the American College of Cardiology*, vol. 76, no. 25, pp. 2982–3021, 2020.
- [4] World Health Organization. ‘Cardiovascular diseases (CVDs)’. (2021), [Online]. Available: [https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)).
- [5] H. D. White, R. M. Norris, M. A. Brown *et al.*, ‘Left ventricular end-systolic volume as the major determinant of survival after recovery from myocardial infarction.’, *Circulation*, vol. 76, no. 1, pp. 44–51, 1987.
- [6] R. M. Norris, H. D. White, D. B. Cross *et al.*, ‘Prognosis after recovery from myocardial infarction: The relative importance of cardiac dilatation and coronary stenoses’, *European heart journal*, vol. 13, no. 12, pp. 1611–1618, 1992.
- [7] O. Bernard, A. Lalande, C. Zotti *et al.*, ‘Deep learning techniques for automatic MRI cardiac multi-structures segmentation and diagnosis: is the problem solved?’, *IEEE transactions on medical imaging*, vol. 37, no. 11, pp. 2514–2525, 2018.
- [8] V. Fuster, J. Narula, P. Vaishnava *et al.*, *Fuster and Hurst’s The Heart, 15e*. New York, NY: McGraw-Hill Education, 2022.
- [9] V. Tavakoli and A. A. Amini, ‘A survey of shaped-based registration and segmentation techniques for cardiac images’, *Computer Vision and Image Understanding*, vol. 117, no. 9, pp. 966–989, 2013.
- [10] C. Chen, C. Qin, H. Qiu *et al.*, ‘Deep learning for cardiac image segmentation: A review’, *Frontiers in Cardiovascular Medicine*, vol. 7, p. 25, 2020, ISSN: 2297-055X. DOI: [10.3389/fcvm.2020.00025](https://doi.org/10.3389/fcvm.2020.00025).
- [11] P. Radau, Y. Lu, K. Connelly *et al.*, ‘Evaluation framework for algorithms segmenting short axis cardiac mri.’, *The MIDAS Journal*, 2009.
- [12] A. Suinesiaputra, B. R. Cowan, A. O. Al-Agamy *et al.*, ‘A collaborative resource to build consensus for automated left ventricular segmentation of cardiac mr images’, *Medical image analysis*, vol. 18, no. 1, pp. 50–62, 2014.

- [13] V. M. Campello, P. Gkontra, C. Izquierdo *et al.*, ‘Multi-Centre, Multi-Vendor and Multi-Disease Cardiac Segmentation: the M&Ms Challenge’, *IEEE Transactions on Medical Imaging*, vol. 40, no. 12, pp. 3543–3554, 2021.
- [14] L. M. Biga, S. Dawson, A. Harwell *et al.*, *Anatomy & physiology*. OpenStax/Oregon State University, 2020.
- [15] Wikimedia Commons, *Diagram_of_the_human_heart_(cropped).svg*, [Online; accessed 08-September-2023], 2023. [Online]. Available: [https://commons.wikimedia.org/wiki/File:Diagram_of_the_human_heart_\(cropped\).svg](https://commons.wikimedia.org/wiki/File:Diagram_of_the_human_heart_(cropped).svg).
- [16] J. Feger, A. Murphy and D. Bell, *Cine imaging (MRI)*, [Online; accessed 08-September-2023], 2023. [Online]. Available: <https://radiopaedia.org/articles/cine-imaging-mri>.
- [17] O. Ronneberger, P. Fischer and T. Brox, ‘U-Net: Convolutional Networks for Biomedical Image Segmentation’, no. arXiv:1505.04597, 2015, arXiv:1505.04597 [cs]. DOI: [10.48550/arXiv.1505.04597](https://doi.org/10.48550/arXiv.1505.04597). [Online]. Available: <http://arxiv.org/abs/1505.04597>.
- [18] E. Kerfoot, J. Clough, I. Oksuz *et al.*, ‘Left-ventricle quantification using residual u-net’, in *Statistical Atlases and Computational Models of the Heart. Atrial Segmentation and LV Quantification Challenges: 9th International Workshop, STACOM 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers 9*, Springer, 2019, pp. 371–380.
- [19] O. Oktay, J. Schlemper, L. L. Folgoc *et al.*, ‘Attention U-Net: Learning where to look for the pancreas. arXiv 2018’, *arXiv preprint arXiv:1804.03999*, 1804.
- [20] A. Hatamizadeh, Y. Tang, V. Nath *et al.*, ‘UNETR: Transformers for 3D Medical Image Segmentation’, in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2022, pp. 574–584.
- [21] F. Milletari, N. Navab and S.-A. Ahmadi, ‘V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation’, in *2016 fourth international conference on 3D vision (3DV)*, Ieee, 2016, pp. 565–571.
- [22] N. Siddique, S. Paheding, C. P. Elkin *et al.*, ‘U-net and its variants for medical image segmentation: A review of theory and applications’, *IEEE Access*, vol. 9, pp. 82 031–82 057, 2021. DOI: [10.1109/ACCESS.2021.3086020](https://doi.org/10.1109/ACCESS.2021.3086020).
- [23] J. Peng and Y. Wang, ‘Medical Image Segmentation with Limited Supervision: A Review of Deep Network Models’, no. arXiv:2103.00429, 2021, arXiv:2103.00429 [cs]. DOI: [10.48550/arXiv.2103.00429](https://doi.org/10.48550/arXiv.2103.00429). [Online]. Available: <http://arxiv.org/abs/2103.00429>.
- [24] W. Bai, O. Oktay, M. Sinclair *et al.*, ‘Semi-supervised learning for network-based cardiac MR image segmentation’, in *Medical Image Computing and Computer-Assisted Intervention- MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11-13, 2017, Proceedings, Part II 20*, Springer, 2017, pp. 253–260.

-
- [25] C. Sudlow, J. Gallacher, N. Allen *et al.*, ‘UK biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age’, *PLoS medicine*, vol. 12, no. 3, e1001779, 2015.
- [26] W. Bai, H. Suzuki, C. Qin *et al.*, ‘Recurrent neural networks for aortic image sequence segmentation with sparse annotations’, no. arXiv:1808.00273, 2018, arXiv:1808.00273 [cs]. DOI: [10.48550/arXiv.1808.00273](https://doi.org/10.48550/arXiv.1808.00273). [Online]. Available: <http://arxiv.org/abs/1808.00273>.
- [27] D. Rueckert, L. Sonoda, C. Hayes *et al.*, ‘Nonrigid registration using free-form deformations: application to breast MR images’, *IEEE Transactions on Medical Imaging*, vol. 18, no. 8, pp. 712–721, 1999, ISSN: 1558-254X. DOI: [10.1109/42.796284](https://doi.org/10.1109/42.796284).
- [28] A. Bitarafan, M. Nikdan and M. S. Baghshah, ‘3D Image Segmentation With Sparse Annotation by Self-Training and Internal Registration’, *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 7, pp. 2665–2672, 2021, ISSN: 2168-2208. DOI: [10.1109/JBHI.2020.3038847](https://doi.org/10.1109/JBHI.2020.3038847).
- [29] K. Chaitanya, E. Erdil, N. Karani *et al.*, ‘Contrastive learning of global and local features for medical image segmentation with limited annotations’, no. arXiv:2006.10511, 2020, arXiv:2006.10511 [cs, eess, stat]. DOI: [10.48550/arXiv.2006.10511](https://doi.org/10.48550/arXiv.2006.10511). [Online]. Available: <http://arxiv.org/abs/2006.10511>.
- [30] T. Chen, S. Kornblith, M. Norouzi *et al.*, ‘A Simple Framework for Contrastive Learning of Visual Representations’, no. arXiv:2002.05709, 2020, arXiv:2002.05709 [cs, stat]. DOI: [10.48550/arXiv.2002.05709](https://doi.org/10.48550/arXiv.2002.05709). [Online]. Available: <http://arxiv.org/abs/2002.05709>.
- [31] D. Zeng, Y. Wu, X. Hu *et al.*, ‘Positional Contrastive Learning for Volumetric Medical Image Segmentation’, no. arXiv:2106.09157, 2021, arXiv:2106.09157 [cs]. DOI: [10.48550/arXiv.2106.09157](https://doi.org/10.48550/arXiv.2106.09157). [Online]. Available: <http://arxiv.org/abs/2106.09157>.
- [32] C. You, W. Dai, F. Liu *et al.*, ‘Mine your own anatomy: Revisiting medical image segmentation with extremely limited labels’, no. arXiv:2209.13476, 2023, arXiv:2209.13476 [cs, eess]. DOI: [10.48550/arXiv.2209.13476](https://doi.org/10.48550/arXiv.2209.13476). [Online]. Available: <http://arxiv.org/abs/2209.13476>.
- [33] A. Tejankar, S. A. Koohpayegani, V. Pillai *et al.*, ‘ISD: Self-Supervised Learning by Iterative Similarity Distillation’, no. arXiv:2012.09259, 2021, arXiv:2012.09259 [cs]. DOI: [10.48550/arXiv.2012.09259](https://doi.org/10.48550/arXiv.2012.09259). [Online]. Available: <http://arxiv.org/abs/2012.09259>.
- [34] R. Bommasani, D. A. Hudson, E. Adeli *et al.*, ‘On the Opportunities and Risks of Foundation Models’, no. arXiv:2108.07258, 2022, arXiv:2108.07258 [cs]. DOI: [10.48550/arXiv.2108.07258](https://doi.org/10.48550/arXiv.2108.07258). [Online]. Available: <http://arxiv.org/abs/2108.07258>.
- [35] J. Devlin, M.-W. Chang, K. Lee *et al.*, ‘Bert: Pre-training of deep bidirectional transformers for language understanding’, *arXiv preprint arXiv:1810.04805*, 2018.
- [36] A. Radford, J. Wu, R. Child *et al.*, ‘Language models are unsupervised multitask learners’, 2019.
-

- [37] A. Radford, J. W. Kim, C. Hallacy *et al.*, ‘Learning transferable visual models from natural language supervision’, no. arXiv:2103.00020, 2021, arXiv:2103.00020 [cs]. DOI: [10.48550/arXiv.2103.00020](https://doi.org/10.48550/arXiv.2103.00020). [Online]. Available: <http://arxiv.org/abs/2103.00020>.
- [38] A. Kirillov, E. Mintun, N. Ravi *et al.*, ‘Segment anything’, no. arXiv:2304.02643, 2023, arXiv:2304.02643 [cs]. DOI: [10.48550/arXiv.2304.02643](https://doi.org/10.48550/arXiv.2304.02643). [Online]. Available: <http://arxiv.org/abs/2304.02643>.
- [39] D. Cheng, Z. Qin, Z. Jiang *et al.*, ‘SAM on Medical Images: A Comprehensive Study on Three Prompt Modes’, no. arXiv:2305.00035, 2023, arXiv:2305.00035 [cs]. [Online]. Available: <http://arxiv.org/abs/2305.00035>.
- [40] Y. Huang, X. Yang, L. Liu *et al.*, ‘Segment Anything Model for Medical Images?’, no. arXiv:2304.14660, 2023, arXiv:2304.14660 [cs, eess]. [Online]. Available: <http://arxiv.org/abs/2304.14660>.
- [41] S. He, R. Bao, J. Li *et al.*, ‘Computer-Vision Benchmark Segment-Anything Model (SAM) in Medical Images: Accuracy in 12 Datasets’, no. arXiv:2304.09324, 2023, arXiv:2304.09324 [cs, eess]. [Online]. Available: <http://arxiv.org/abs/2304.09324>.
- [42] J. Ma, Y. He, F. Li *et al.*, ‘Segment Anything in Medical Images’, no. arXiv:2304.12306, 2023, arXiv:2304.12306 [cs, eess]. [Online]. Available: <http://arxiv.org/abs/2304.12306>.
- [43] F. Isensee, P. F. Jaeger, S. A. Kohl *et al.*, ‘nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation’, *Nature methods*, vol. 18, no. 2, pp. 203–211, 2021.
- [44] F. Hutter, L. Kotthoff and J. Vanschoren, *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.
- [45] DeepMind, *surface-distance*, [Online; accessed 20-September-2023], 2023. [Online]. Available: <https://github.com/google-deepmind/surface-distance>.
- [46] M. J. Cardoso, W. Li, R. Brown *et al.*, ‘MONAI: An open-source framework for deep learning in healthcare’, Nov. 2022. DOI: <https://doi.org/10.48550/arXiv.2211.02701>.
- [47] A. Paszke, S. Gross, F. Massa *et al.*, ‘PyTorch: An Imperative Style, High-Performance Deep Learning Library’, in *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: <http://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.