Knowledge Distillation for Reducing User Input Burden in Interactive Medical Image Segmentation

Dhruv Rachakonda
University of Utah

UUCS-25-001

School of Computing
University of Utah
Salt Lake City, UT 84112 USA

16 April 2025

Abstract

Medical image segmentation is a critical task in the healthcare domain, aiding in the precise delineation of anatomical structures. However, this process is often challenging due to the fine-grained yet low-detail nature of medical images, making automated segmentation difficult. To improve segmentation accuracy, many models incorporate interactive user input, such as points, scribbles, and bounding boxes. While this interaction enhances performance, it introduces a significant user burden, sometimes requiring up to 50 clicks per image to achieve high accuracy. This demand on annotators, particularly in large-scale medical datasets, presents a major barrier to efficient image labeling. To address this challenge, we propose a new knowledge distillation mode, with the intention of reducing user interaction while maintaining high segmentation accuracy. Unlike existing knowledge distillation methods that primarily focus on compressing image encoders or improving model efficiency for resource-constrained environments, our approach leverages distillation specifically to minimize the need for extensive user input. By reducing annotation burden, our approach can enhance the efficiency of medical image labeling software and alleviate the workload of anatomy professionals handling large-scale datasets.

KNOWLEDGE DISTILLATION FOR REDUCING USER INPUT BURDEN IN INTERACTIVE MEDICAL IMAGE SEGMENTATION

by

Dhruv Varahavenkatasai Rachakonda

A Senior Honors Thesis Submitted to the Faculty of

The University of Utah

In Partial Fulfillment of the Requirements for the

Honors Degree in Bachelor of Science

In

Data Science

Approved:

Dr. Shireen Elhabian
Thesis Faculty Advisor

Dr. Mary Hall
Chair, Kalhert School of Computing

Dr. Thomas Henderson
Honors Faculty Advisor

April 2025

ABSTRACT

Medical image segmentation is a critical task in the healthcare domain, aiding in the precise delineation of anatomical structures. However, this process is often challenging due to the fine-grained yet low-detail nature of medical images, making automated segmentation difficult. To improve segmentation accuracy, many models incorporate interactive user input, such as points, scribbles, and bounding boxes. While this interaction enhances performance, it introduces a significant user burden, sometimes requiring up to 50 clicks per image to achieve high accuracy. This demand on annotators, particularly in large-scale medical datasets, presents a major barrier to efficient image labeling. To address this challenge, we propose a new knowledge distillation mode, with the intention of reducing user interaction while maintaining high segmentation accuracy. Unlike existing knowledge distillation methods that primarily focus on compressing image encoders or improving model efficiency for resource-constrained environments, our approach leverages distillation specifically to minimize the need for extensive user input. By reducing annotation burden, our approach can enhance the efficiency of medical image labeling software and alleviate the workload of anatomy professionals handling large-scale datasets.

Dedicated to my parents

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Within the field of modern healthcare and medical imaging, medical image segmentation plays a crucial role, enabling automated analysis and decision-making within areas such as radiology, pathology, and other fields. Accurate segmentation of medical images, such as the ones that are obtained from medical devices such as CT Scans, X-rays, and MRI machines, is essential for detecting abnormalities, planning treatments, conducting large scale medical studies, and especially for generating large scale annotated datasets for future data analytics and machine learning purposes. For example, figure 1 shows an example segmentation of an axial slice of the brain segmented into different components. However, despite many advances in the field of medical image segmentation, many methods still require the use of user interaction to refine and advance image segmentation results.

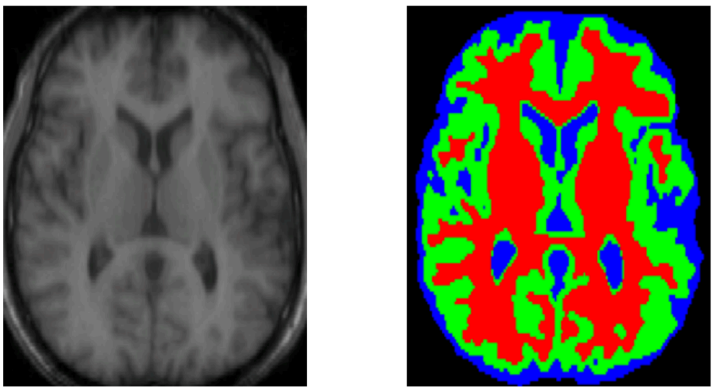


Figure 1: Tissue Segmentation of Axial Slice in Brain

Interactive image segmentation involves the use of user-guided annotations, such as clicks, scribbles, and bounding boxes to assist segmentation models differentiate between

regions of interest and background areas. This process, while very effective at improving accuracy, imposes a significant user burden on users that utilize medical image segmentation software, who must manually provide precise user inputs for large datasets. Certain medical image segmentation models, in order to achieve high accuracy, can require many user-clicked points and bounding boxes over the course of multiple iterations of user inputs **[3,6]** . Other methods require the use of live-wire methods, which requires users to manually place points along object boundaries to guide segmentation **[4,5]** . Reducing this input burden without also reducing the segmentation accuracy and effectiveness remains a critical challenge in the field of medical image segmentation.

In order to combat this user input burden, and bridge this research gap, we turn to the technique of knowledge distillation. This is a technique where smaller, more efficient models learn from a larger, pre-trained model. Existing research in knowledge distillation techniques for image segmentation has primarily focused on compressing computationally intensive models for resource constrained environments **[1]**. However, research revolving around the application of knowledge distillation specifically for reducing user-input burden is limited.

To bridge this gap, we propose a novel feature-level knowledge distillation model designed with the objective of reducing user input in interactive medical image segmentation. Unlike most traditional approaches that require extensive user inputs **[3,4,5,6]**, our model enables accurate segmentation with fewer user interactions while maintaining performance comparable to models with higher input requirements, having potential to improve the efficiency of medical annotations and alleviate the burdensome process of labeling large-scale medical datasets.

The remainder of this thesis is as follows. In Chapter 2, we discuss the background of interactive segmentation and knowledge distillation in medical imaging. In Chapter 3, we review

existing work such as existing knowledge distillation approaches and interactive segmentation models. In Chapter 4, we introduce our model, detailing its architecture and methodology. In Chapter 5, we show our experimental results and comparative findings on our model's performance. Finally, in Chapter 6, we conclude the thesis with a discussion of our results and findings, limitations, and future implications and research based on our findings.

CHAPTER 2

Background and Motivation

2.1 Challenges of Medical Image Segmentation

Despite advancements in technology, several challenges persist in this domain, impacting the accuracy and efficiency of segmentation methods. One major concern is how to consistently acquire high-quality images that can provide reliable and interpretable information for disease diagnosis and treatment **[13,17]**. Medical images are often captured under a variety of conditions, such as different lighting levels and capturing distances, which can lead to inconsistencies across datasets. In many cases, the resulting images have extremely poor resolution, making it difficult to detect and diagnose lesions, especially when they are small or located in complex anatomical regions. Additionally, clinical images may include artifacts such as hair, shadows, and reflections, which can obscure relevant features and hinder accurate lesion discrimination and analysis **[13,18]**.

These inconsistencies are further exacerbated by variations in brightness, lighting, and the presence of noise, all of which hinder the performance of automated segmentation models. For instance, dermoscopic or endoscopic images often suffer from visual obstructions like hair and specular highlights, which reduce segmentation accuracy **[13,15]**. In other cases, such as CT imaging, noise, blur, and low contrast are common due to the intrinsic nature of X-ray acquisition and the use of lower radiation doses aimed at minimizing patient risk. While lower doses help reduce radiation exposure, they also lead to reduced image quality, making it more difficult to identify subtle pathological features **[13,14]**. For example, figure 2 shows an example of a brainstem medical image segmentation, demonstrating the graininess and poorly lit nature of

medical images in general. These limitations not only compromise the visual clarity of medical images but also pose significant challenges for both manual interpretation by clinicians and automated segmentation by machine learning models.

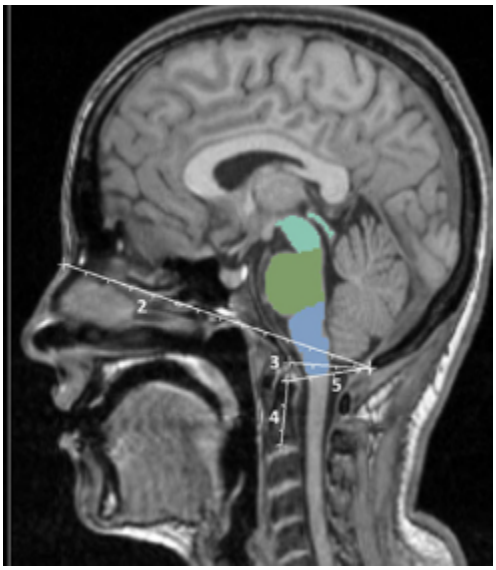


Figure 2: Example Image Segmentation of a Brainstem Medical Image

To address the persistent challenges in medical image segmentation, such as inconsistent quality, low contrast, and the presence of visual artifacts, user interaction has become a valuable complement to automated methods. Rather than relying solely on fully automated systems, which may struggle under these conditions, many approaches incorporate forms of human guidance to improve segmentation accuracy **[3,5,6,7]**. By using inputs such as clicks, bounding boxes, or scribbles, users can provide contextual cues that help models better localize and delineate structures of interest. This collaborative process leverages both computational efficiency and expert knowledge, offering a practical pathway to more reliable segmentation outcomes in clinical settings **[5,6,7]**.

2.2 User Interaction in Medical Image Segmentation

In order to help combat the persistent challenges in medical image segmentation, much ongoing research turns to the use of user input, such as point clicks, bounding boxes, and scribbles **[3, 19]** These interactive techniques allow for real-time corrections and guidance, addressing the limitations of fully automated systems, particularly when segmenting regions with ambiguous boundaries or heterogeneous textures.

One particular model, known as ScribblePrompt, demonstrates the effectiveness of these results. ScribblePrompt is an interactive segmentation tool designed to assist in biomedical image analysis by allowing users to delineate structures using scribbles, clicks, and bounding boxes **[20]**. During its experimentation and testing, ScribblePrompt reduced annotation time by 28% and improved segmentation accuracy by 15% compared to the next best method **[20]**. These enhancements are attributed to the tool's design, which incorporates user interactions to refine segmentation results, enabling precise adjustments to the identified structures. Figure 3 shows how ScribblePrompt's two different variants compare to other models, demonstrating its effectiveness. This approach not only accelerates the annotation process but also enhances the accuracy of segmentations, demonstrating the significant impact of user input in improving medical image analysis.

	MedScribble (n=93)		ACDC (n=2,130)	
Model	↑ Dice Score	↓ HD95	↑ Dice Score	↓ HD95
SAM (ViT-b)	0.40 ± 0.05	32.58 ± 3.80	0.20 ± 0.01	108.39 ± 0.85
SAM (ViT-h)	0.56 ± 0.05	14.61 ± 3.18	0.42 ± 0.02	59.78 ± 2.02
SAM-Med2D w.o. adapter	0.52 ± 0.05	30.34 ± 3.29	0.16 ± 0.01	107.25 ± 1.12
SAM-Med2D w/ adapter	0.30 ± 0.06	19.04 ± 3.63	0.17 ± 0.02	41.01 ± 3.04
MIDeepSeg	0.68 ± 0.04	6.94 ± 1.06	0.58 ± 0.01	5.78 ± 0.26
MIDeepSeg w/ refinement	0.81 ± 0.03	3.10 ± 0.67	0.73 ± 0.01	2.71 ± 0.18
MedSAM (box)	0.70 ± 0.04	7.54 ± 1.35	0.70 ± 0.02	3.53 ± 0.28
ScribblePrompt-SAM	0.87 ± 0.02	2.61 ± 0.84	0.77 ± 0.01	4.17 ± 0.32
ScribblePrompt-UNet	0.84 ± 0.02	2.92 ± 0.88	0.84 ± 0.01	1.80 ± 0.11

Figure 3: ScribblePrompt Results from the Paper, Demonstrating the Effectiveness of User Input. See Chapter 5 for explanation on the evaluation metrics, such as Dice Score and HD95

Another model, which is the backbone of our knowledge distillation model (See Section 2.4), is the PRISM Model. PRISM (Promptable and Robust Interactive Segmentation Model) is another model, similar to ScribblePrompt, designed for precise segmentation of 3D medical images, allowing various user inputs such as points, boxes, scribbles, and masks **[3]**. Its architecture is built upon the principle of iterative learning, where the model progressively refines segmentations using previous prompts. Validated across four public datasets focusing on tumor segmentation in the colon, pancreas, liver, and kidney, PRISM demonstrated significant performance improvements over existing methods, achieving results approaching human-level accuracy **[3]**. The incorporation of diverse user interactions enables the model to iteratively enhance segmentation precision, underscoring the critical role of user input in refining interactive medical image analysis. Figure 4 shows how effective PRISM's model is compared to other popular models.

Methods	Public datastes (Dice % / NSD %)			
	Colon tumor	Pancreas tumor	Liver tumor	Kidney tumor
nnU-Net [12]	43.91 / 52.52	41.65 / 62.54	60.10 / 75.41	73.07 / 77.47
3D UX-Net [15]	28.50 / 32.73	34.83 / 52.56	45.54 / 60.67	57.59 / 58.55
Swin-UNETR [28]	35.21 / 42.94	40.57 / 60.05	50.26 / 64.32	65.54 / 72.04
SAM [14]	28.83 / 33.63	24.01 / 26.74	8.56 / 5.97	36.30 / 29.86
3DSAM-adapter [9]	57.32 / 73.65	54.41 / 77.88	56.61 / 69.52	73.78 / 83.86
ProMISe [18]	66.81 / 81.24	57.46 / 79.76	58.78 / 71.52	75.70 / 80.08
SAM-Med3D [31]	54.34 / 78.58	65.61 / 92.40	23.64 / 26.97	76.50 / 88.41
SAM-Med3D-organ	70.75 / 91.03	76.40 / 97.75	66.52 / 77.97	88.20 / 97.80
SAM-Med3D-turbo	73.77 / 94.95	74.87 / 96.43	69.36 / 81.70	89.26 / 98.40
PRISM-plain	67.18 / 85.28	65.73 / 89.51	79.70 / 91.60	85.29 / 93.55
PRISM-ultra	**93.79** / **99.96**	**94.48** / **99.99**	**94.18** / **99.99**	**96.58** / **99.80**

Figure 4: PRISM Results from the Paper, Demonstrating User Input Effectiveness

However, these two models, despite their accuracy, have significant user input requirements. For example, ScribblePrompt requires the user to use scribbles, points, and bounding boxes in order to achieve the high accuracy that is shown in the paper **[20]**. Such types of inputs can present a very heavy user burden, and when such large data sets require the use of scribbling the desired organ, or putting a bounding box around it, this can increase the amount of time that generating large datasets takes **[23].** Figure 5 shows an example of the amount of user input that ScribblePrompt takes in, requiring scribbles and clicks across the image multi

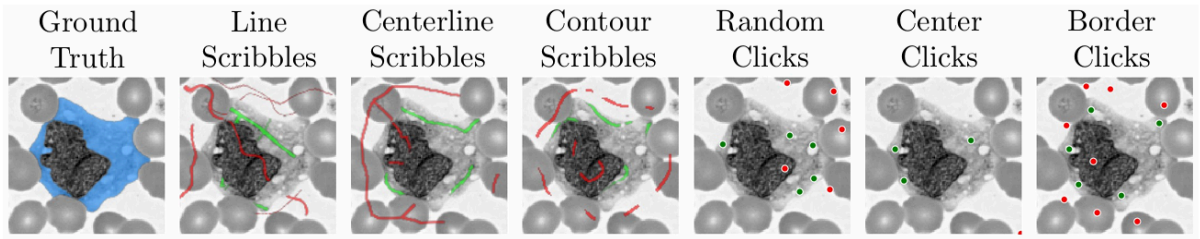


Figure 5: ScribblePrompt's User Input Requirements

Similarly, PRISM has a very significant user-prompt burden. PRISM's results claim to yield dice scores of above 0.9 after 11 iterations of user input **[3]**. However, in order to achieve

its high accuracy, users are required to use its ultra-model **[3]**. The ultra-model requires the use of additional points, bounding boxes, and scribbles, which takes significantly more time due to the lengthy user inputs that it requires **[23]**. If users simply want to utilize one point per image, PRISM-plain can be used, where users are only required to enter one point per image **[3]**. However, these results show no improvement through its iterative process compared to the ultra-model, as shown in figure 6 **[3]** .

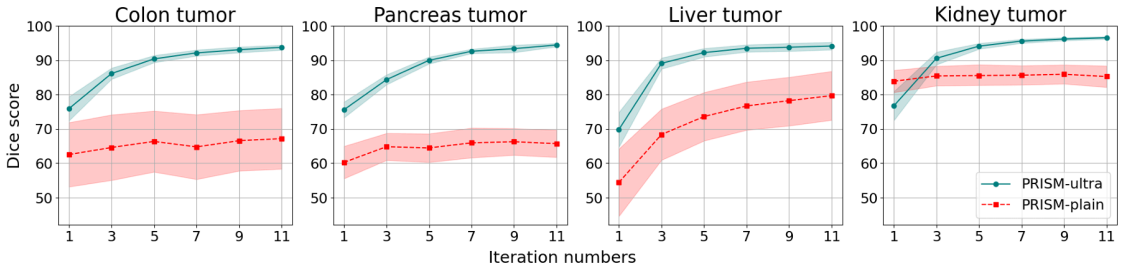


Figure 6: PRISM's Results Demonstrating Poor Performance with Less Burdensome Input

Our objective in this thesis is to explore the potential of knowledge distillation as a tool to reduce user input burden in interactive medical image segmentation. Existing methods often rely on extensive user input, such as detailed clicks or scribbles, to achieve high accuracy, which can be time-consuming and impractical in clinical settings **[3, 19, 20, 23]**. By leveraging knowledge distillation, we aim to transfer knowledge from models trained with higher levels of user interaction to those that require fewer inputs. This approach has the potential to maintain, or even improve, segmentation accuracy while significantly lowering the amount of manual effort required from users. Ultimately, our goal is to make interactive segmentation more efficient and accessible without compromising on performance.

2.3 The VIT Architecture

The primary architecture of many modern image segmentation models, including the one within this thesis, utilizes the Vision Transformer (VIT). The VIT was introduced to address the issue of limited input representation in standard transformer architectures, which were originally designed for sequential data like text, limiting their applicability to images **[8]**. Figure 7 shows an overview of the VIT pipeline, which will be covered in this section.

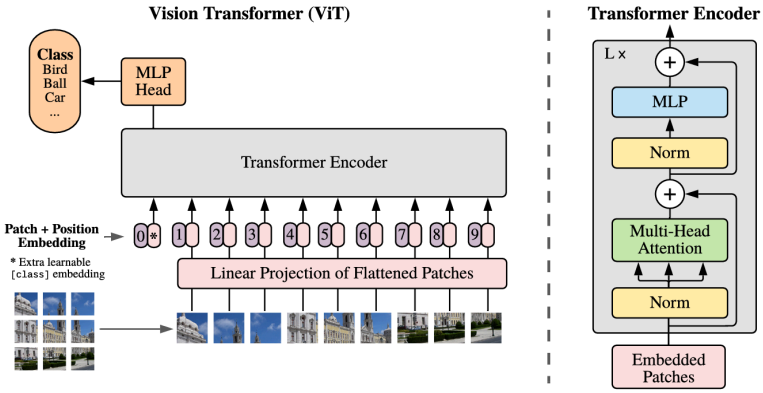


Figure 7: Diagram of the Vision Transformer Flow

In the first step, the image is divided into fixed-size patches. These patches are then transformed into numeric vectors, or embeddings. Embeddings are useful because they map the input data into a vector space where similar data points are placed closer together. This property helps the model recognize patterns and relationships more effectively. Specifically, to embed an image, the ViT divides it into smaller patches (for example, 16x16 pixel tiles). These patches are then flattened and linearly projected into high-dimensional vectors.

Once the image patches are converted into embeddings, they are passed into the first component of the ViT architecture, which is the image encoder. This encoder is typically a Multi-Layer Perceptron (MLP) that processes the embeddings to capture spatial relationships

between the patches. This is similar to how transformers process tokens in text, where the sequential input is crucial to understanding context.

An important aspect of the ViT is the inclusion of positional embeddings. Just as in the standard transformer architecture for natural language, the relative position of each patch within the image is important for understanding the spatial layout. These positional encodings are concatenated to the image embeddings before being input into the image encoder. This step ensures that the model can take into account both the content of the patches and their spatial arrangement within the image **[24]**.

Once the image patches are embedded and the positional encodings are added, the resulting sequence of vectors is passed into the transformer layers. These layers consist of multi-head self-attention mechanisms and feed-forward networks, which are responsible for learning complex relationships and interactions between the patches. The self-attention mechanism allows the ViT to weigh the importance of different patches in relation to each other, enabling it to capture long-range dependencies within the image. This is particularly useful for tasks like image segmentation, where the context of distant regions of the image can be crucial for accurate predictions.

The transformer layers are stacked, and after several layers of self-attention and processing, the output from the final transformer layer is passed through a classifier or segmentation head, depending on the task. For image segmentation, the output tokens corresponding to each image patch are reshaped and upsampled to match the original image size, followed by a pixel-wise classification to generate the segmentation mask. Finally, the output of the segmentation head is compared to the ground truth, and the model is trained using

backpropagation, typically optimizing a loss function like cross-entropy or dice loss, which is designed to minimize the difference between predicted and actual segmentation masks.

2.4 PRISM - The Backbone of Our Knowledge Distillation Model

Our implementation of knowledge distillation is built around PRISM. This section provides an overview of what it is.

PRISM (Promptable and Robust Interactive Segmentation Model) is a framework designed to improve the precision of 3D medical image segmentation through interactive learning **[3]**. It accepts a variety of visual prompts, such as points, boxes, and scribbles as sparse inputs, as well as masks as dense inputs to help guide the segmentation process. Based on the Segment Anything Model (SAM) **[26]**, PRISM operates on four foundational principles. The first principle is iterative learning. By utilizing visual prompts from previous iterations, PRISM refines segmentations progressively, improving accuracy with each cycle. The second principle is confidence learning: Employing multiple segmentation heads per input, each generating a candidate mask accompanied by a confidence score, PRISM optimizes predictions by focusing on the most reliable outputs. The third principle is corrective learning. After each segmentation iteration, a shallow corrective refinement network reassigns mislabeled voxels, enhancing the overall segmentation quality. The final principle is hybrid design. Integrating hybrid encoders, PRISM effectively captures both local and global image features, addressing the complex anatomical variations present in medical images.

PRISM employs a generic encoder-decoder architecture with human-in-the-loop capabilities integrated **[3]**. The image encoder and visual prompt encoder extract latent features from the input image and prompts, respectively. These are fused via self and cross- attention

mechanisms to produce rich embeddings, which are then decoded into a preliminary segmentation output. This output is refined iteratively based on new prompts derived from erroneous regions of previous predictions. The image encoder itself is a hybrid module composed of parallel convolutional neural network (CNN) and vision transformer (VIT) pathways. This design enables PRISM to effectively learn both local anatomical structures and global contextual cues. The decoder output is then used in conjunction with the visual embeddings to generate multiple candidate masks, each assigned a confidence score via lightweight multi-layer perceptrons (MLPs). A selector module identifies the highest-confidence prediction, which is subsequently refined by a corrective refinement network composed of two residual blocks. This network takes a four-channel input: the original image, the selected binary mask, and the cumulative positive and negative prompt maps, as shown in figure 8.

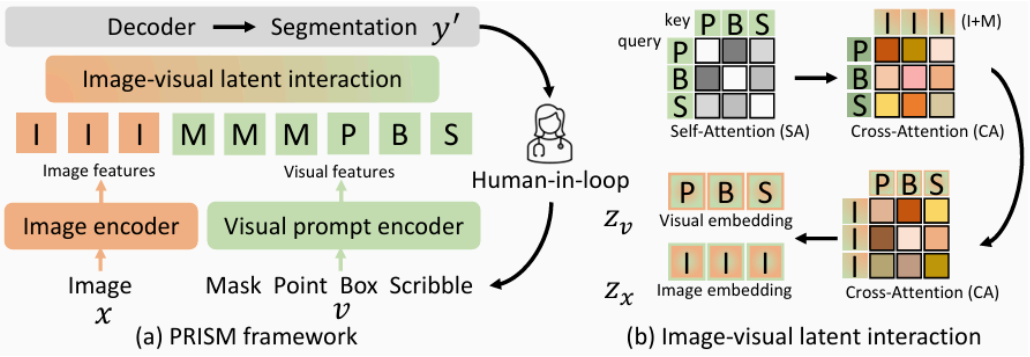


Figure 8: PRISM Architecture Diagram. PRISM will take an image and the visual prompts associated with it such as points, bounding boxes, and scribbles. Then, the user will provide prompts for the next iteration. In part (b), the model combines features from the image and the user's prompts by letting them interact in a shared space. It uses special attention mechanisms to help the model focus on important details from both sources and create meaningful representations for the image and prompts.

Prompts are automatically sampled from regions of disagreement between the predicted and ground truth segmentations, simulating human corrections. Sparse prompts (points and scribbles) are derived through structured sampling and deformation strategies, while a fixed 3D bounding box is used throughout the iterations. Additionally, the dense prompt for any iteration is derived from the logits map of the previous iteration, allowing gradient flow and preserving contextual learning. By integrating these modules, PRISM not only enables flexible prompt integration but also facilitates progressive refinement toward expert-level segmentation performance.

However, as mentioned in section 2.2, PRISM's human in the loop feature imposes a significant burden on the user. PRISM's plain model, which only requires the use of points during the iterative process, performs poorly over the course of 11 iterations compared to its ultra counterpart, which requires the use of multiple points, bounding boxes, and scribbles. Our objective in this thesis is to try and outperform the PRISM plain model, while still using the same amount of points as the plain model.

2.5 Knowledge Distillation

This section provides an introduction to knowledge distillation, and an overview of how it functions.

Knowledge distillation is essentially an optimization technique where a larger, high intensity model, transfers knowledge to a less accurate model **[25]**. This method allows these low performing models to increase in accuracy due to this new transfer of knowledge while still

maintaining the low latency required for this model to perform. This larger, high intensity model, is known as the teacher model whereas the weaker model is known as the student student model.

To simplify the process of explaining knowledge distillation, we will utilize an example of a simple neural network. Generally, the first instinct when it comes to where the knowledge is stored is to assume that it is stored within the weights and biases since this is where the results of training are. However, since the objective of knowledge distillation is to compress this network, compressing the weights and biases matrix would be very difficult to see. Rather, we look at the network as a function that transforms an input into an output vector. For an image classification task, we have an image that is transformed into a probability distribution, usually with an output function such as a softmax operation. The knowledge, in this case, is the probability distribution at the output level. Figure 9 shows an example of a neural network output displayed as a probability distribution when classifying the MNIST dataset.

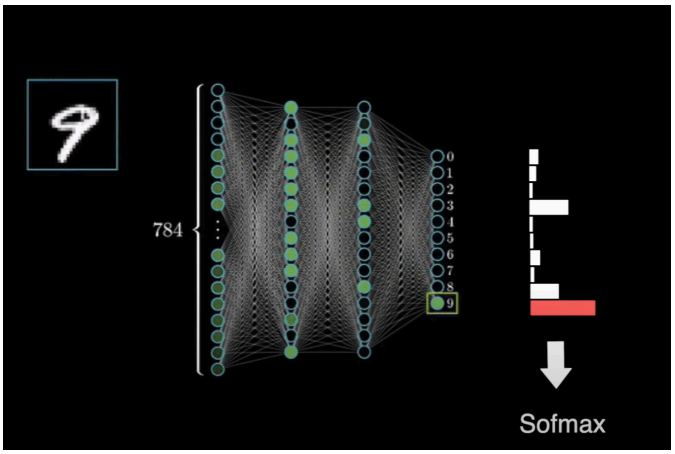


Figure 9: An example of the classic MNIST classification task. However, here we have put the output in terms of a probability distribution. This essentially shows that even though the answer is 9, the distribution shows similarities to 8, or 3, which also shows that knowledge can be carried in terms of a probability distribution.

The specific variant of knowledge distillation that our thesis uses is called Soft Distillation [**26**]. Soft distillation revolves around the idea of measuring the distance between two probability functions. The primary metric that ongoing research utilizes is Kullback-Leibler Divergence (KL Divergence), which is defined as follows

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x)\ \log\left(\frac{P(x)}{Q(x)}\right)$$

where *P* and are two probability functions whose distance is being measured. The KL Divergence is then factored into the loss function during the training process based on the objective function, or the loss function being utilized (eg. MSE, Cross Entropy). In the case of our knowledge distillation problem, the functions P and Q represent the student and teacher models' softmax outputs. The following loss function is an example of KL Divergence factored into a standard cross entropy loss function. Though the this loss function is not the one used in our model, it provides an overview of how knowledge distillation can be implemented into a standard loss function, such as cross entropy.

$$\mathcal{L}_{\text{global}} = (1-\lambda)\mathcal{L}_{\text{CE}}(\psi(\mathbf{Z}_s), y) + \lambda\tau^2 \text{KL}(\psi(\mathbf{Z}_s/\tau), \psi(\mathbf{Z}_t/\tau))$$

In the equation, we also have added a new constant, τ. This is the temperature constant that is used when determining how sensitive the probability distribution will be when generating our softmax outputs. During knowledge distillation, we utilize the softmax with temperature activation function written below.

$$\text{Softmax}_\tau(z_i) = \frac{\exp(z_i/\tau)}{\sum_j \exp(z_j/\tau)}$$

The factor, τ, essentially dictates how "sensitive" the probability distribution is. A higher value for τ means that larger numbers will output far higher probabilities from the softmax function the larger the value for τ is. Naturally, the opposite occurs for numbers that are small. A smaller value for τ means that the probabilities will be far less sensitive. A large number inputted into the softmax function won't have as high a probability as a large τ, and it is vice versa for smaller numbers. Once we have obtained our loss function in terms of the distance between the teacher and the student distributions, we can now train our model using this new loss function.

CHAPTER 3

Related Work

3.1 Summary of Existing and Ongoing Research

Ongoing research in knowledge distillation primarily revolves around reducing latency, and allowing large models to work in resource constrained environments **[1, 28, 29]**. Though a majority of this research revolves around improving the accuracy of lightweight models, there is no existing work regarding the use of knowledge distillation to improve lightweight models, where lightweight in this case means models with low user-input burden. In this chapter, we will go over three currently existing models that utilize feature distillation as their optimization technique to better establish the ongoing research in the field.

3.2 Classic Knowledge Distillation Techniques

Traditional knowledge distillation methods **[25, 38]** involve transferring the output probability distribution from a teacher model to a student model using soft targets. These methods typically optimize KL Divergence between teacher and student outputs. Extensions of this idea have been explored in natural language processing, image classification, and object detection, with temperature scaling and soft targets playing a critical role in improving student learning. However, these techniques generally operate at the output level, neglecting intermediate feature representations that can be especially valuable in dense prediction tasks like segmentation.

3.3 Feature-Level Knowledge Distillation in Segmentation

Recent research has shifted toward distillation at the feature or embedding level to improve fine-grained spatial understanding, particularly in tasks like semantic segmentation and medical imaging **[22, 29]**. These approaches use losses like mean squared error (MSE) or cosine similarity to align intermediate representations from the teacher and student. Models such as MobileSAM and TransKD **[1, 28]** have demonstrated that distilling encoder or attention block features can improve performance in lightweight networks. Some works also incorporate architectural adaptations, such as projection heads or self-attention fusion, to bridge dimensional mismatches between teacher and student features **[7, 32].** Our proposed model draws inspiration from these works but diverges by applying feature-level supervision specifically to interactive segmentation with the goal of reducing manual user input, rather than working in computational resource starved environments.

3.4 Interactive Segmentation and User Input Dependency

In the context of medical imaging, interactive segmentation models like ScribblePrompt **[20]** and PRISM **[3]** have demonstrated strong performance using user-provided clicks, scribbles, or bounding boxes. These models iteratively refine segmentation outputs by incorporating user corrections. However, their reliance on dense, repeated user input presents a bottleneck in terms of annotator time and scalability. While effective, they do not attempt to alleviate this burden using learning-based techniques. Our work fills this gap by proposing a distillation mechanism that allows a sparse-input student model to benefit from a richer-input teacher model, thereby maintaining performance while reducing interaction overhead.

Chapter 4

Implementation and Methodology

4.1 Teacher and Student Architecture

Our distillation model starts with a teacher and student network, both based on a generic encoder and decoder architecture. Borrowing from the PRISM Architecture, our model takes an input image ***x***, along with the visual prompt ***v,*** the image and prompt encoders generate the image and prompt embeddings **[3]**. These embeddings are then fed into the resulting decoder architecture to produce the final image segmentation *y'*. The image encoder, specifically, is a hybrid architecture consisting of a ViT **[8]** and a Convolutional Neural Network (CNN).

Adapted on SAM **[8, 27, 30]** for 3D medical images, as well their visual prompts including sparse prompts (such as bounding boxes, scribbles, and point clicks) and dense prompts (such as segmentation masks), the prompt encoder utilizes the SAM framework to generate embeddings for the user prompt. The same principle, with a 3D adaptation, applies to the mask decoder as well. Figure 10 shows the pipeline of the bothe the teacher and student models borrows from PRISM.

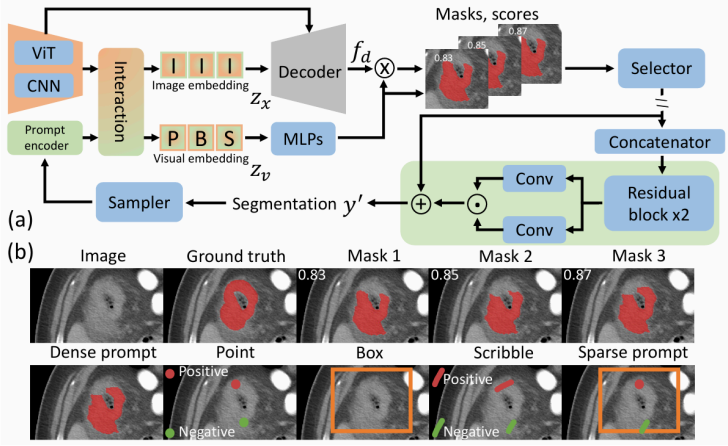


Figure 10: PRISM Architecture used for our Teacher and Student Models

4.2 Teacher Model

The first part of our architecture is the teacher network. The teacher network is built upon the PRISM "ultra-plus" model, since it is a strong performing model according to experiments done with simulating user interaction **[3]**. For the purposes of this overview, the "ultra-plus" model will be referred to as the "teacher" model, since it is the backbone of the teacher model in this knowledge distillation model. The teacher model is trained on a dynamic number of point clicks, ranging from 1 to up to 50 points. The number of points randomized during each simulated iteration. Additionally, the teacher model is also trained using bounding boxes, as well as scribbles. For the use of our overall architecture, the teacher is a pre-trained network, and is frozen during the knowledge distillation training.

4.3 Student Model

The next part of our architecture is the student network. The student network is built upon the PRISM "plain-b-1" model. For the purposes of this overview, the "plain-b-1" model will be referred to as the "student" model, since it is the backbone of the student model in this knowledge distillation model. Normally, the plain-b-1 model uses a bounding box during the training process in each iteration. However, in order to reduce user prompt burden as well as provide better supervision and give multiple locations of interest, we have modified it to remove the need of a bounding box. Additionally, the student model previously used 1 point in the PRISM model; we now use five points to provide better supervision as well.

The student model and the teacher model both use the same architecture shown in figure 10. The only differing factor between both the teacher and the student models lies in the amount of input that is required.

4.4 Distillation Pipeline

With our teacher and student models' architecture established, we can now move onto the process of how knowledge is "distilled" from the teacher to the student. Specifically, our model distills information at the feature level. Though logit-level distillation is more common **[25, 29]**, we have opted to use feature-level distillation since KD neglects intermediate-level supervision for complete guidance **[31].** What this means is that since we have multiple components, such as the multiple encoders and decoders, these building blocks constitute an embedding level meaning and a distillation at each individual component is coherent. This allows flexibility in modifying the student architecture based on specific needs. For example, MobileSAM uses feature level decoupled distillation **[28],** where their research has shown an increase in performance in Dice Scores, as well as a reduction in training requirements.

Distillation occurs at multiple components of our architecture. Specifically, it occurs at the visual prompt embedding, image prompt embedding, and the mask decoder embedding level. The first part of our model is the teacher prompt encoder. The visual prompt is first fed into the teacher prompt encoder, where prompt embeddings are generated. A subset of visual prompts that are low effort, such as a few points clicks, is also fed into the student encoder as well. We now want to distill the rich representation of the teacher model into the student model. However, this cannot be trivially done because there is a dimensionally mismatch between the teacher and student embeddings. This mismatch occurs due to the teacher model being prompted with a

richer set of user-inputs, such as bounding boxes and scribbles, as well as many point clicks. Similarly, the student model has lower dimensionality since there are far less points embedded into the student prompt embeddings.

In order to reduce the teacher embeddings to the same dimensionality as the student embeddings, we utilize a linear projection head **[7, 32].** The linear projection head helps to encode the information to align in dimensionality with respect to the student embeddings. The projection simply derives from $Z_T W_P$ where Z_T is the teacher model, and W_P is a learnable matrix representing the linear projection. The distillation objective here is now to minimize the embedding difference between the projected teacher, and the student embeddings. To do so, we utilize the Mean-Square Error (MSE) loss function and minimize it. Because PRISM is an iterative framework **[3],** the segmentation mask from the previous iteration is passed on as a dense mask prompt back into prompt encoder, as shown in figure 11.

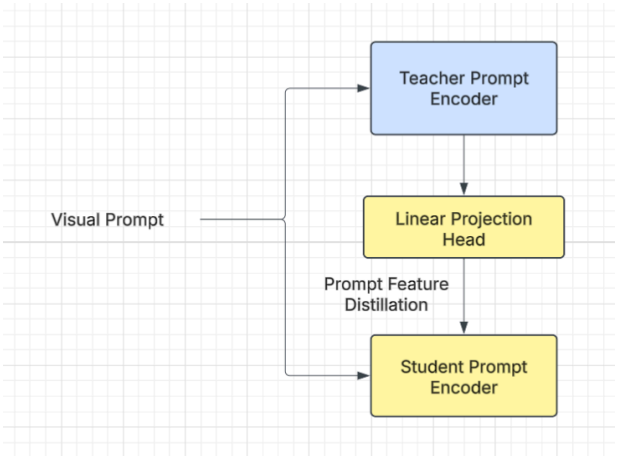


Figure 11: Visual Prompt Encoder Distillation from Teacher to Student Using Projection

The next part of our model is the distillation within the image encoder, as shown in figure 12. The teacher image encoder, which is a frozen pre-trained model, is inferred with an image

input and the embeddings that are produced from the image encoder are then distilled to the student image encoder to generate student image embeddings. Unlike the prompt encoders, the distillation process for the image encoders does not require the use of a projection head since the teacher and student image encoders are the same architecture, as well as taking in the same image input. This results in the same embedding dimensions for both the teacher and student image embeddings.

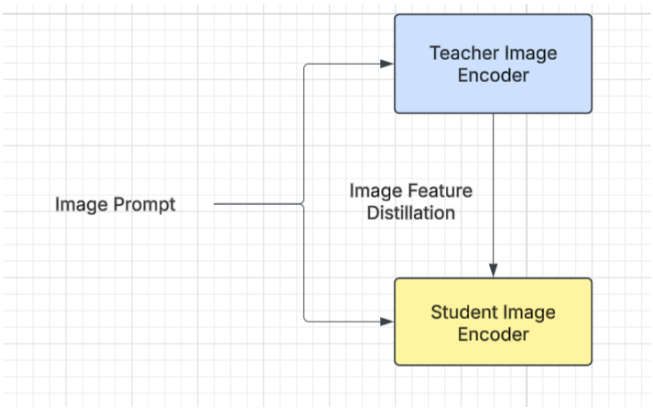


Figure 12, Image Prompt Encoder Distillation from Teacher to Student

The next part of our model is the distillation process between teacher and student image and prompt interaction. The image and prompt embeddings from the respective teacher and student model are fed into a two way transformer **[27]** interaction module where a cross attention mechanism from prompt to image, and image to prompt interactions are carried out and generates contextualized image and prompt embeddings. We then perform distillation between these contextualized embeddings. This process is diagrammed in figure 13.

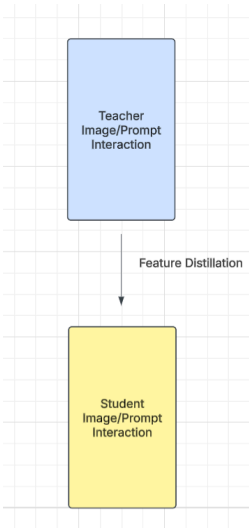


Figure 13: Two Way Transformer Distillation Process for Prompt and Image Interaction

Finally, our model ends at the mask decoders for both the teacher and the student, where both models take in the contextualized image and prompt embeddings from the two way transformer and generate segmentation masks for both the teacher and student. We then perform distillation at the output logit level for these segmentation masks. We now have a student mask that has been generated as the final output segmentation, as shown in figure 14.

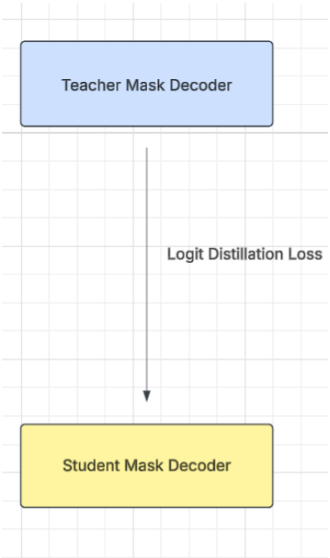


Figure 14: Mask Decoder Distillation Process for Segmentation Masks

The entirety of the distillation process is diagrammed in figure 15. This is simply the distillation pipeline which classified the various pixels using our methodology. The training is a different process, and is discussed in section 4.5.

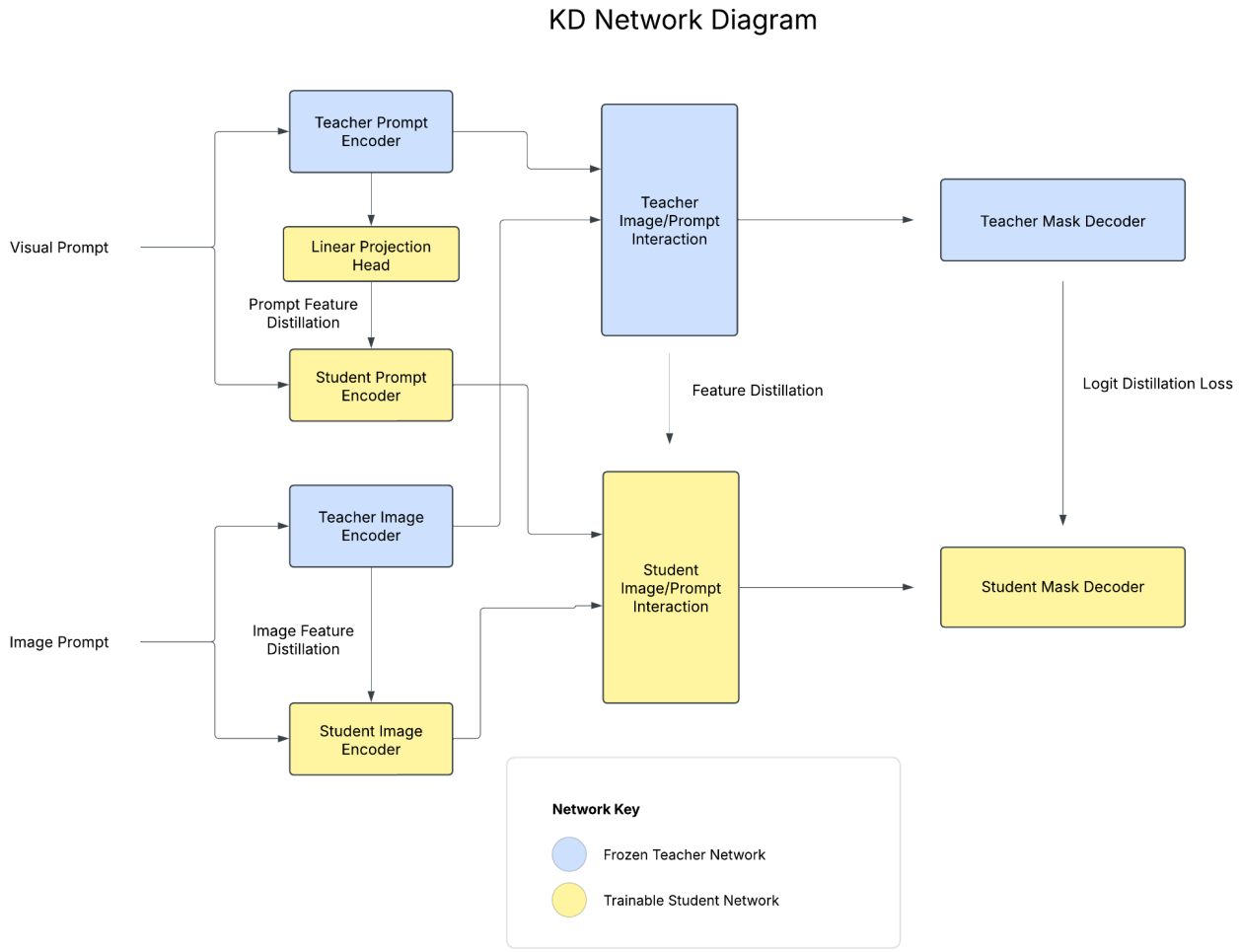


Figure 15: Overall KD Network Diagram With All Components

4.5 Distillation Training Process

In order to train our model, we utilize a combination of relative MSE loss as well as KL Divergence. Relative MSE is calculated at the prompt encoder and image encoder embeddings to determine the deviation between student and teacher probability distributions. KL Divergence is calculated at the output logit level distillation to determine the ditfference between teacher and student level segmentation masks. The probability of this distribution is calculated using Softmax with Temperature. The overall objective of our model is to collectively minimize the prompt encoder-image encoder MSE loss, as well as the KL Divergence between the teacher and student output segmentation mask distribution

First, we need to construct the distillation loss function for the distillation that occurs between the teacher image embedding and the student image embedding. For this, we can simply utilize the Mean Squared Error between the teacher and student image embedding

$$L_{distil-image} = MSE(teacher\_image\_embedding, student\_image\_embedding)$$

Then, we need to construct the distillation loss function for the distillation that occurs between the teacher prompt embedding, as well as the teacher student embedding. Similarly to the distillation between the image embeddings, we can utilize the Mean Squared Error between the teacher and student prompt embeddings

$$L_{distil-prompt} = MSE(teacher\_prompt\_embedding, student\_prompt\_embedding)$$

Now, we need to establish the distillation loss between the teacher and student mask decoders' output logits. Unlike the image and prompt embeddings, which were vectors, the

output logits can be fed into a softmax function with temperature constant. This results in a probability distribution for all of the outputs. Therefore, we can utilize the KL Divergence between the probability distributions

$$L_{distill-logit} = KLD(teacher\_mask, student\_mask)$$

Now, in order to calculate the total distillation loss, we simply add up all of the distillation losses of the prompt and image encoders, as well as the logit distillation loss.

$$L_{distil} = L_{distill-prompt} + L_{distill-image} + L_{distill-logit}$$

The segmentation loss is based on the backbone PRISM model, and does not change for this loss function, adopting the confidence learning framework for the loss function **[3].**

$$L_{seg} = L_{corrective} + L_{confidence}$$

We have now established all of the building blocks for our final training loss function. We add up the total loss over all the iterations during the iterative learning process. We can now construct our final loss function below as follows

$$L_{training} = \sum_{i=1}^{N} L_{seg} + \alpha L_{Distill}$$

This loss function is essentially the segmentation loss plus the distillation loss added together. This sum is then summed over the course of the interactions done in PRISM. However, it is important to note a new hyperparameter that has been added, α; this is referred to as the alpha constant. The alpha constant here is a parameter that is multiplied by the distillation loss to determine how much "importance" the distillation loss is to provide to the overall loss **[33]**. A

large value for alpha means that the distillation losses contribute more to the training loss, whereas a low value for alpha means that distillation plays a lower importance in determining the overall training loss. In chapter 5, we explore different values for alpha to determine how it impacts the performance on our knowledge distillation.

In addition to the alpha hyperparameter, we modify the loss functions through a series of experiments by implementing log scaling, as well as using the relative MSE loss function. The details of these modifications are discussed in Chapter 5.

CHAPTER 5

Experimentation - Performance and Results

In this chapter, we discuss the experiments we conducted in order to evaluate the performance of our knowledge distillation model, as well as an overview of the results of these experiments. The main objective of our experiments is to test our results with different parameters, such as the alpha value discussed in chapter 4.

5.1 Dataset

The dataset that we have chosen to use for our series of experiments is the Medical Segmentation Decathlon (MDS) dataset **[34]**. Specifically, we have opted to use the Task 10 - Colon part of the dataset. Figure 16 shows an example segmentation from another model which utilizes the MDS Task 10 - Colon dataset.

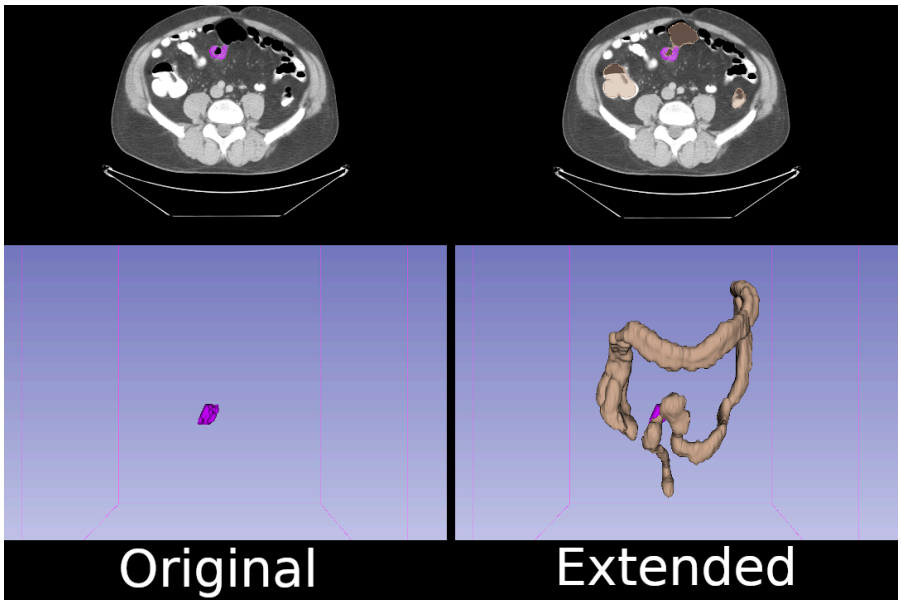


Figure 16: Example Segmentation from the Task 10 - Colon Dataset

In addition to this dataset, the research team of PRISM has also created a pre-processed dataset of the colon CT scan images **[3]**. The dataset, having been pre-processed, has automatically been split into their respective training sets. As part of our experimentation, we will use this pre-processed dataset.

5.2 Evaluation Metrics

As part of our experimentation, we utilize the Dice-Sørensen Score, or the Dice Score metric in order to determine how our model performs. The Dice Score is calculated as follows

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|}$$

where $|X|$ and $|Y|$ are the sizes of the predicted and ground truth segmentation masks (i.e correct segmentation masks), respectively. In the context of image segmentation, this corresponds to the number of pixels in the predicted region (X) and the ground truth region (Y). The intersection $|X \cap Y|$ represents the number of pixels correctly identified by the model.

The Dice Score ranges from 0 to 1, with 1 indicating perfect overlap between the predicted segmentation and the ground truth, and 0 indicating no overlap at all. This metric is especially useful in evaluating segmentation performance when dealing with imbalanced data or small structures, as it emphasizes the overlap between the predicted and actual regions.

Ongoing research uses other metrics, along with Dice Scores. For example, as shown in figure 3, ScribblePrompt uses the Hausdorff distance, or HD95 Distance. The HD95 distance measures how far two shapes are from each other; in this case, it measures how far the boundaries of the segmentation are from each other. Formally, this is defined as

$$d_{\mathrm{H}}(X, Y) := \max \left\{ \sup_{x \in X} d(x, Y), \ \sup_{y \in Y} d(X, y) \right\}$$

For two sets X and Y in the image space. However, we did not use the HD95 distance in our experiments because our focus was on overall region overlap rather than boundary outliers, which the Dice Score captures more directly and consistently.

Image segmentation evaluation can also involve the use of normalized surface distance, or NSD. For example, as shown in figure 4, PRISM utilizes NSD as one of their evaluation metrics. In the context of image segmentation, NSD measures how well the boundaries of the predicted segmentation match the ground truth, within a certain tolerance. Formally, this is defined as

$$\mathrm{NSD}_\tau = \frac{|\{x \in S_P : d(x, S_G) \leq \tau\} \cup \{y \in S_G : d(y, S_P) \leq \tau\}|}{|S_P| + |S_G|}$$

where S_P is the set of surface points of the predicted segmentation, S_G is the set of surface points of the ground truth segmentation, $d(x, S_G)$ the shortest distance from point x to the surface S_G, τ is the distance tolerance (e.g., 1–2 pixels or mm, depending on context). We did not use NSD in our experiments because it focuses on boundary accuracy within a set tolerance, whereas our primary goal, as mentioned before, was to assess overall region overlap, which the Dice Score captures more directly and is simpler to interpret and compute.

5.3 Experimentation Overview

For our experiments, our student model utilizes only 5 points for the user input. The teacher model, which distills information to the student, uses an arbitrary number of points, as well as bounding boxes, and scribbles.

As part of our experiments, we have set up a few hyperparameters. The first one is the alpha constant, as discussed in chapter 4 to determine the "importance" of the distillation loss on the overall loss. The second one is the option to use relative MSE, instead of MSE in the loss functions for the distillation of the prompt. Relative MSE is simply the MSE loss, divided by the L_2 norm of the teacher embedding vectors. The third one is the option to include a log scaling factor, which essentially scales down all the values of the distillation loss by feeding it into a logarithm. This would allow very large values that may occur in the distillation process to be scaled back in order to better match the segmentation loss, within the overall loss function. Overall, we conduct a series of 6 experiments.

In the graphs that follow, the validation scores are shown in a line graph, whereas the test scores are shown in a box plot. Below are the experiments that we conducted.

Experiment	Alpha	MSE Loss	Log Scaling
1	2.0	Standard	None
2	50.0	Standard	None
3	100.0	Standard	None
4	500.0	Standard	None
5	2.0	Relative	None
6	5.0	Relative	Enabled

5.4 Experimentation Results

Baseline Teacher Model: Points, Bounding Boxes, and Scribbles (Upper Bound)

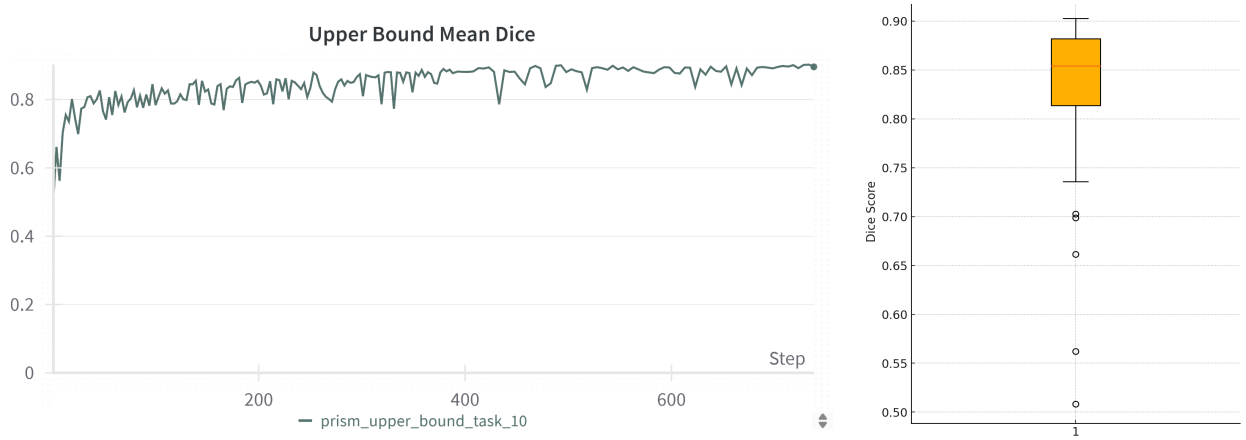


Figure 17: Baseline Teacher Model with Input Heavy Dice Scores

Baseline PRISM Model: Only 5 Points, No Distillation:

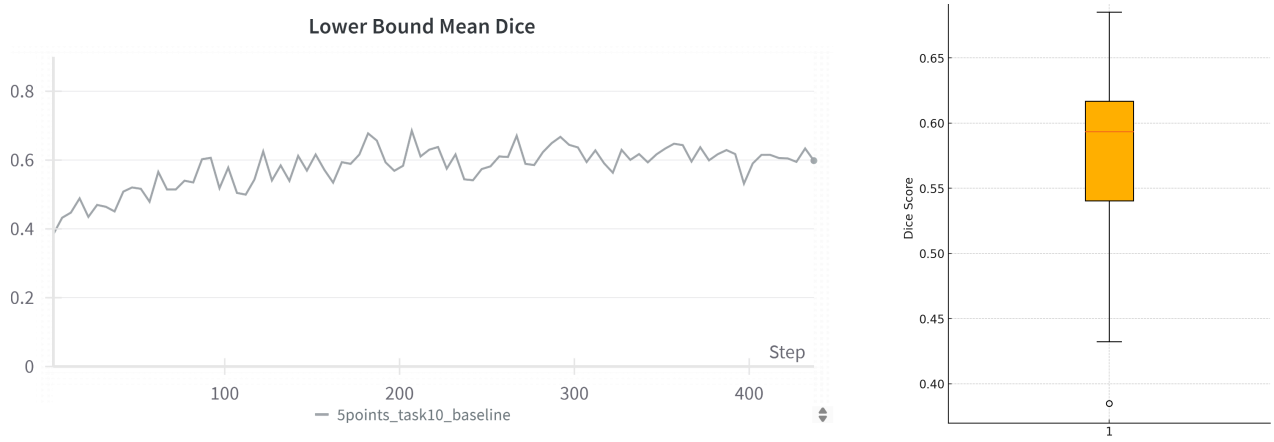


Figure 18: Baseline PRISM Model with Reduced User Input and No Distillation

With our lower bound baseline model established, as well as the upper bound teacher model as well, we can now test our student model versus the baseline lower bound. Our objective here is to outperform the lower bound despite the lack of user input.

Experiment 1: Alpha 2.0, Standard MSE

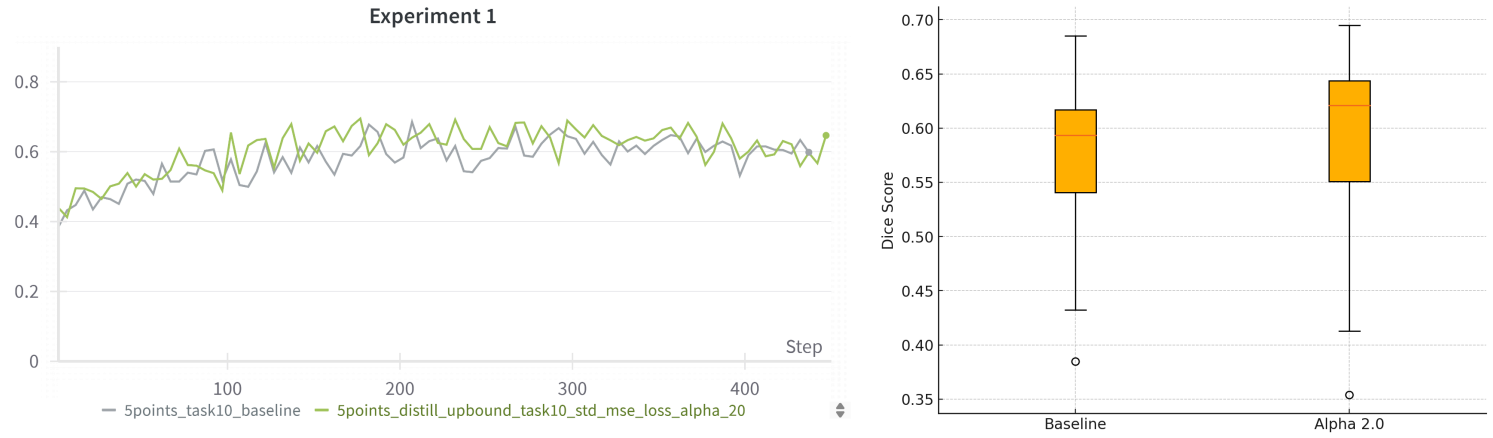


Figure 19: Experiment 1 Results with Standard MSE, Alpha 2.0

Experiment 2: Alpha 50.0, Standard MSE

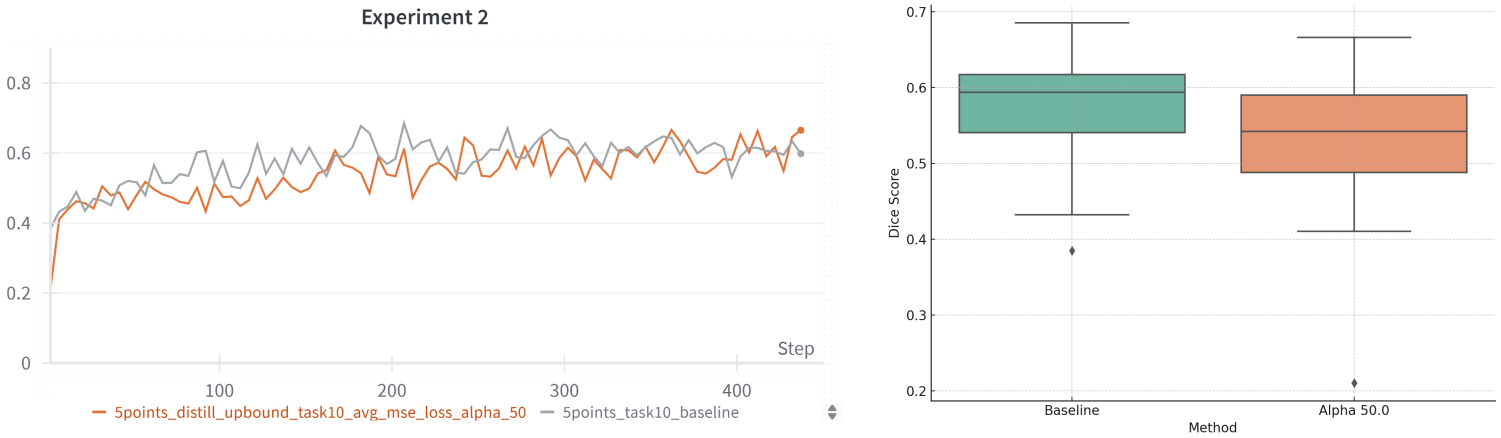
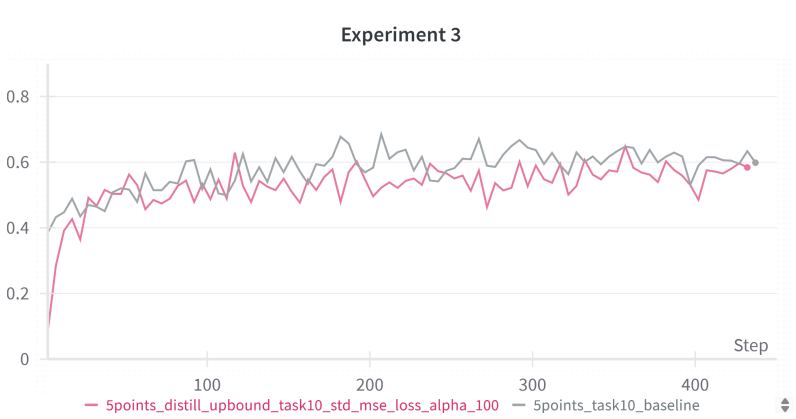


Figure 20: Experiment 2 Results with Standard MSE, Alpha 50.0

Experiment 3: Alpha 100.0, Standard MSE



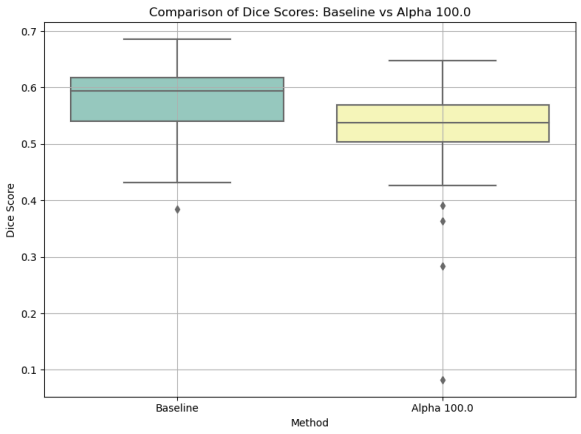
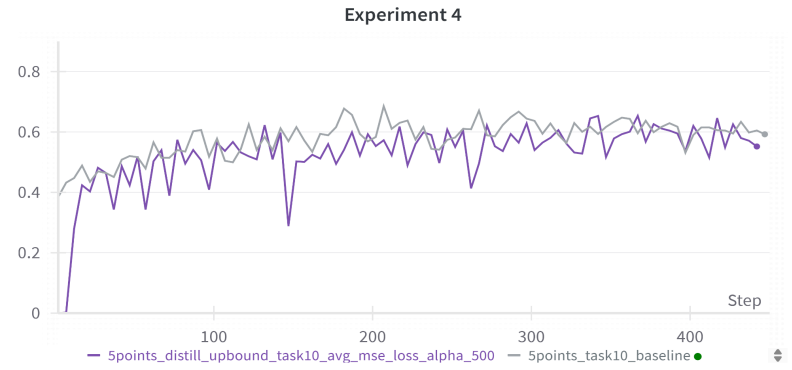


Figure 21: Experiment 3 Results with Standard MSE, Alpha 100.0

Experiment 4: Alpha 500.0, Standard MSE



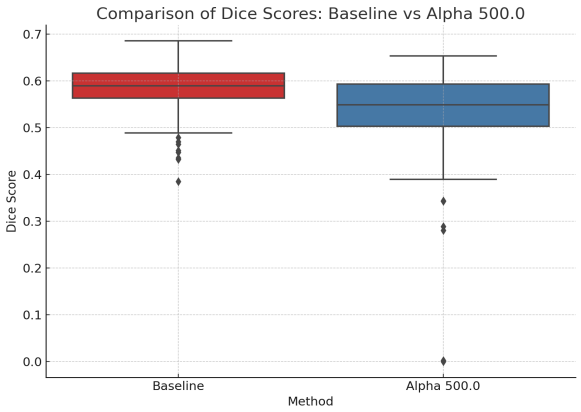


Figure 22: Experiment 4 Results with Standard MSE, Alpha 500.0

Experiment 5: Alpha 2.0, Relative MSE

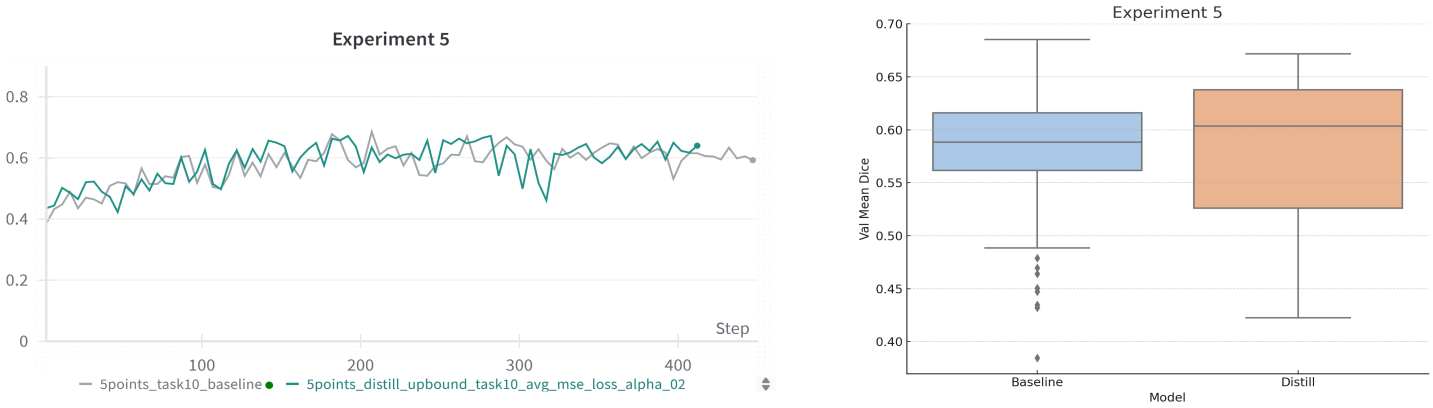


Figure 23: Experiment 5 Results with Relative MSE, Alpha 2.0

Experiment 6: Alpha 5.0, Relative MSE, Log Scaling Enabled

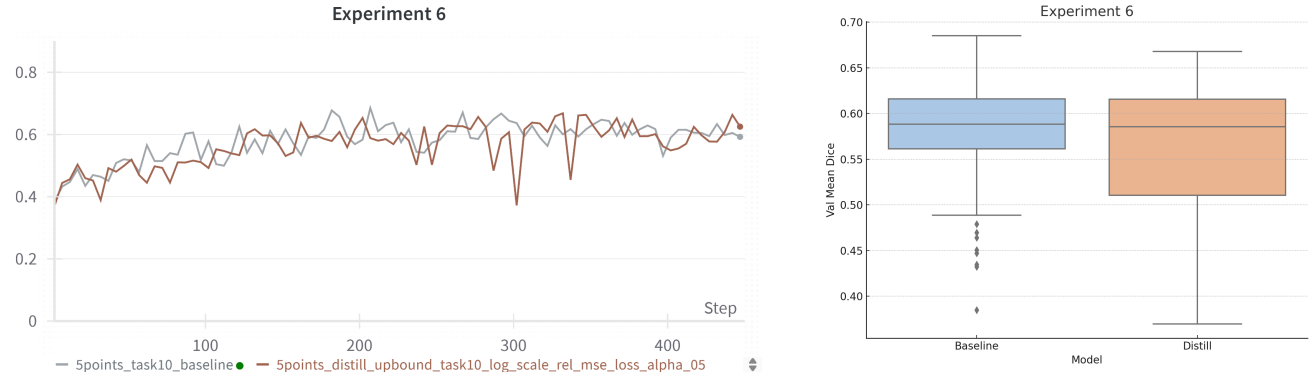


Figure 24: Experiment 6 Results with Relative MSE, Log Scaling Enabled, and Alpha 5.0

The results of these experiments show promising results in some experiments. In other experiments, however, the results are not conclusive. We discuss the results of these experiments in the next chapter.

CHAPTER 6

Discussion

6.1 Results

The results of our experiments demonstrate that feature-level knowledge distillation can meaningfully reduce user input burden in interactive medical image segmentation without significantly compromising performance. Compared to the PRISM "plain" baseline model, our student model, with only five point-based inputs and no bounding boxes or scribbles, was able to match or outperform the baseline in certain configurations, suggesting that distillation from a high-performing teacher model can potentially compensate for reduced user prompts.

Among all experiments, the models that incorporated relative MSE loss (Experiments 1 and 5) showed the most promising performance. This aligns with prior findings in distillation literature , where normalizing the loss with respect to the teacher's embedding magnitudes helps stabilize training and reduce the impact of dimensionality disparities between teacher and student **[35]**. Experiment 5, which combined relative MSE with log scaling and a modest alpha value (alpha = 5.0), yielded the most balanced outcome in terms of training stability and final Dice scores. This suggests that controlling the scale of the distillation loss is essential when integrating it with standard segmentation losses.

Interestingly, experiments with higher alpha values (Experiments 2-4), which increased the weight of the distillation loss in training, did not perform as well. In fact, extremely high alpha values (e.g., alpha = 500 in Experiment 4) resulted in diminished test performance, implying that overly emphasizing distillation can distract the student from learning from the direct segmentation loss. These results highlight the importance of balancing the importance of

distilled knowledge with the loss function, especially in iterative frameworks like PRISM where segmentation precision improves through interactions

Our findings also validate the hypothesis that feature-level distillation is especially advantageous for architectures involving multi-stage encoders and prompt-based interactions. Unlike logit-level distillation, which only affects the final output, feature-level distillation enriches the intermediate representations that guide segmentation decisions throughout the network **[22]**. This is particularly critical in interactive frameworks like PRISM, where prompt and image embeddings are fused iteratively across multiple transformer blocks.

Some inconsistencies in performance across experiments point to areas for further refinement (See chapter 7). While knowledge distillation successfully reduced the user input burden, the performance was sensitive to hyperparameters such as alpha, projection head design, and temperature scaling. This indicates that a one-size-fits-all distillation setup may not be optimal; future work could benefit from strategies that tune distillation importance dynamically based on feedback from segmentation accuracy during training.

These experiments show the potential of knowledge distillation not only as a tool for model compression as ongoing research shows **[1,7,11,22,26,29]** but also as a mechanism for reducing annotation effort in clinical workflows. By enabling accurate segmentations with limited input, such frameworks can streamline dataset generation and improve the accessibility of AI-assisted medical imaging tools in resource-constrained settings. With further experimentation and fine tuning, validation across diverse medical imaging datasets, this approach could be a step toward more user-efficient, high-performance interactive segmentation systems.

CHAPTER 7

Conclusion

7.1 Conclusion

In this thesis, we presented a framework leveraging feature-level knowledge distillation to reduce user input burden in interactive medical image segmentation. By transferring rich intermediate representations from a high-performing teacher model (trained with dense user inputs such as points, bounding boxes, and scribbles) to a student model (trained on only five sparse point inputs), we demonstrated the viability of maintaining segmentation performance while dramatically reducing the manual effort required during annotation.

Our methodology introduced a distillation pipeline that performed embedding-level supervision at multiple stages: visual prompt encoders, image encoders, and mask decoders. We experimented with a variety of hyperparameters, including standard and relative MSE, KL divergence on soft logits, and log scaling mechanisms. These experiments provided meaningful insights into the optimal balance between segmentation loss and distillation loss, revealing that moderate distillation weights combined with normalized embedding loss yield the most stable and effective results.

The results of our experiments, particularly in Experiments 1 and 5, indicate that our distilled student model outperforms the baseline PRISM plain model, which uses the same number of inputs but lacks teacher supervision. This suggests that our approach has the potential to meaningfully reduce the user interaction overhead without compromising performance, offering a promising step forward in the development of more efficient medical annotation tools.

This work bridges the gap between high-performance segmentation and practical usability by incorporating interactive learning and representation distillation. As the field of medical imaging continues to grow in complexity and scale, approaches like ours, focused on reducing user workload without sacrificing quality, will become increasingly critical in supporting both clinicians and machine learning practitioners.

7.2 Future Work and Implications

While this work presents encouraging results in reducing user input for interactive segmentation through feature-level knowledge distillation, there remain several places researchers can explore for further improvement. One way is the expansion of this framework to multiple medical dataset types and diverse imaging modalities beyond the colon CT scans used in this study. Applying the same distillation strategy to datasets involving MRI, ultrasound, or multi-modal scans would help evaluate the model's robustness and generalizability across clinical use cases.

Additionally, our current experiments used a fixed number of point prompts (5) for the student model. Future work could explore flexible input schemes, where the model randomly determines the minimal number of prompts required per image based on requirements. This would push the boundaries of input efficiency even further by minimizing interaction without hardcoding limits. Further research could hopefully even bring the number of points required down to merely one point.

There is also potential in enhancing the distillation architecture itself. While our model used linear projection heads to align the feature dimensions between teacher and student, more advanced techniques like attention-guided distillation **[36]**, multi-scale feature fusion **[37]**, or teacher projection reuse **[7]** could offer richer supervision and more meaningful feature transfer.

With further improvements, the proposed knowledge distillation strategy could serve as a foundation for next-generation segmentation systems, capable of learning from rich supervision while operating efficiently in real-world, user-driven environments.

7.3 Limitations

As mentioned in the future work section, a key limitation lies in the scope and scale of evaluation. All experiments were conducted using only the Task 10 (Colon) dataset from the Medical Segmentation Decathlon **[34]**. While this dataset provides a controlled testbed for evaluating segmentation performance, it does not reflect the full spectrum of complexity seen across different organs, pathologies, or imaging modalities.

Moreover, user interaction was simulated rather than sourced from real human annotators, meaning that factors like annotation inconsistency, fatigue, or spatial bias were not accounted for. Future work will need to validate the effectiveness of the proposed approach in real-world annotation pipelines, with clinical experts providing input in-the-loop. This will be essential to truly assess the framework's impact on user burden and its practicality in medical environments.

Additionally, the model's performance and stability are highly sensitive to several hyperparameters, such as the alpha weight for distillation loss, the choice between standard versus relative MSE, and the inclusion of log scaling. As demonstrated in the experimental results, improper tuning of these factors can lead to unstable convergence or diminished segmentation accuracy. This dependence introduces a level of fragility and necessitates extensive empirical experimentation to find an optimal configuration, which may not be feasible in real-world deployment scenarios

REFERENCES

[1] Liu, R., Yang, K., Roitberg, A., Zhang, J., Peng, K., Liu, H., Wang, Y., & Stiefelhagen, R. (2023). *TransKD: Transformer Knowledge Distillation for Efficient Semantic Segmentation.* arXiv. https://doi.org/10.48550/arXiv.2202.13393

[2] Deshpande, T., Prakash, E., Ross, E. G., Langlotz, C., Ng, A., & Valanarasu, J. M. J. (2024). *Auto-Generating Weak Labels for Real & Synthetic Data to Improve Label-Scarce Medical Image Segmentation.* arXiv. https://doi.org/10.48550/arXiv.2404.17033

[3] Li, H., Liu, H., Hu, D., Wang, J., & Oguz, I. (2024). *PRISM: A Promptable and Robust Interactive Segmentation Model with Visual Prompts.* arXiv. https://doi.org/10.48550/arXiv.2404.15028

[4] Grady L. *Minimal surfaces extend shortest path segmentation methods to 3d.* IEEE Trans Pattern Analysis Mach Intell 2008;32:321–34.

[5] Poon M, Hamarneh G, Abugharbieh R. *Efficient interactive 3d livewire segmentation of complex objects with arbitrary topology*. Comput Med Imaging Graph 2008;32:639–50.z

[6] Liang X, Shen X, Feng J, Lin L, Yan S. *Semantic object parsing with graph lstm.* European conference on computer vision 2016:125–43.

[7] Nguyen, K.-B., & Park, C. J. (2024). Retro: *Reusing Teacher Projection Head for Efficient Embedding Distillation on Lightweight Models via Self-Supervised Learning*. arXiv. https://doi.org/10.48550/arXiv.2405.15311

[8] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.* arXiv. https://doi.org/10.48550/arXiv.2010.11929

[9] Thisanke, H., Deshan, C., Chamith, K., Seneviratne, S., Vidanaarachchi, R., & Herath, D. (2023). *Semantic segmentation using Vision Transformers: A survey*. arXiv. https://doi.org/10.48550/arXiv.2305.03273

[10] Qin, D., Bu, J.-J., Liu, Z., Shen, X., Zhou, S., Gu, J.-J., Wang, Z.-H., Wu, L., & Dai, H.-F. (2021). *Efficient medical image segmentation based on knowledge distillation*. *arXiv*. https://doi.org/10.48550/arXiv.2108.09987

[11] Wang, S., Yan, Z., Zhang, D., Wei, H., Li, Z., & Li, R. (2023). *Prototype knowledge distillation for medical segmentation with missing modality*. *arXiv*. https://doi.org/10.48550/arXiv.2303.09830

[12] Zhang, C., Han, D., Zheng, S., Choi, J., Kim, T.-H., & Hong, C. S. (2023). *MobileSAMv2: Faster Segment Anything to Everything*. *arXiv*. https://doi.org/10.48550/arXiv.2312.09579

[13] Deheyab, A. O. A., Alwan, M. H., Abdul Rezzaqe, I. K., Mahmood, O. A., Hammadi, Y. I., Kareem, A. N., & Ibrahim, M. (2022). *An overview of challenges in medical image processing. Proceedings of the 6th International Conference on Future Networks & Distributed Systems (ICFNDS '22)*, 1–8. https://doi.org/10.1145/3584202.3584278

[14] Al-Ameen, Z., & Sulong, G. *Prevalent Degradations and Processing Challenges of Computed Tomography Medical Images:* A Compendious Analysis. International Journal of Grid and Distributed Computing, vol.9(10), pp.107-118, 2016.

[15] Chakravorty, R., Liang, S., Abedini, M., & Garnavi, R. "Dermatologist-like feature extraction from skin lesion for improved asymmetry classification in PH 2 database". In Engineering in Medicine and Biology Society (EMBC), IEEE 38th Annual International Conference of the , pp. 3855-3858, 2016.

[16] Yang F, Li X, Duan H, Xu F, Huang Y, Zhang X, Long Y, Zheng Y. MRL-Seg: Overcoming Imbalance in Medical Image Segmentation With Multi-Step Reinforcement Learning. IEEE J Biomed Health Inform. 2024 Feb;28(2):858-869. doi: 10.1109/JBHI.2023.3336726. Epub 2024 Feb 5. PMID: 38032774.

[17] Tajbakhsh N, Jeyaseelan L, Li Q, Chiang JN, Wu Z, Ding X. Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. Med Image Anal. 2020 Jul;63:101693. doi: 10.1016/j.media.2020.101693. Epub 2020 Apr 3. PMID: 32289663.

[18] Oliveira, R. B., Marranghello, N., Pereira, A. S., & Tavares, J. M. R. "A computational approach for detecting pigmented skin lesions in macroscopic images". Expert Systems with Applications, vol.61, pp.53-63, 2016.

[19] Xu, W., Liang, Z., Anthony, H., Ibrahim, Y., Cohen, F., Yang, G., Whitehouse, D., Menon, D., Newcombe, V., & Kamnitsas, K. (2024). *Continuous online adaptation driven by user interaction for medical image segmentation*. *arXiv*. https://doi.org/10.48550/arXiv.2503.06717v1

[20] Wong, H. E., Rakic, M., Guttag, J., & Dalca, A. V. (2024*). ScribblePrompt: Fast and flexible interactive segmentation for any biomedical image*. *arXiv*. https://doi.org/10.48550/arXiv.2312.07381

[21] Miles, R., & Mikolajczyk, K. (2024). *Understanding the role of the projector in knowledge distillation*. *arXiv*. https://doi.org/10.48550/arXiv.2303.11098

[22] Heo, B., Kim, J., Yun, S., Park, H., Kwak, N., & Choi, J. Y. (2019). A comprehensive overhaul of feature distillation. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 1921–1930. https://doi.org/10.1109/ICCV.2019.00201

[23] Yanglan Ou, Sharon X. Huang, Kelvin K. Wong, Jonathon Cummock, John Volpi, James Z. Wang, Stephen T.C. Wong, *BBox-Guided Segmentor: Leveraging expert knowledge for accurate stroke lesion segmentation using weakly supervised bounding box prior,*
Computerized Medical Imaging and Graphics,
https://doi.org/10.1016/j.compmedimag.2023.102236.

[24]Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention is all you need. arXiv*.
https://doi.org/10.48550/arXiv.1706.03762

[25] Hinton, G., Vinyals, O., & Dean, J. (2015). *Distilling the Knowledge in a Neural Network.* arXiv preprint arXiv:1503.02531. Retrieved from https://arxiv.org/pdf/1503.02531

[26] Zhang, Y., & Yan, D. (2025). *Soft Knowledge Distillation with Multi-Dimensional Cross-Net Attention for Image Restoration Models Compression*. arXiv preprint arXiv:2501.09321.

[27] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., & Girshick, R. (2023). *Segment Anything.* arXiv preprint arXiv:2304.02643. https://arxiv.org/pdf/2304.02643

[28] Zhang, C., Han, D., Qiao, Y., Kim, J. U., Bae, S.-H., Lee, S., & Hong, C. S. (2023). *Faster Segment Anything: Towards Lightweight SAM for Mobile Applications.* arXiv preprint arXiv:2306.14289.x1x

[29] Zhao, L., Qian, X., Guo, Y., Song, J., Hou, J., & Gong, J. (2023). MSKD: *Structured knowledge distillation for efficient medical image segmentation.* Computers in Biology and Medicine, 164, 107284. https://doi.org/10.1016/j.compbiomed.2023.107284

[30] Wang, H., Guo, S., Ye, J., Deng, Z., Cheng, J., Li, T., Chen, J., Su, Y., Huang, Z., Shen, Y., Fu, B., Zhang, S., He, J., & Qiao, Y. (2023). SAM-Med3D: *Towards general-purpose segmentation models for volumetric medical images*. *arXiv*. https://doi.org/10.48550/arXiv.2310.15161

[31] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, L. G., Schmidt, M., Chen, M., Ravichandran, A., Wright, G., Adam, H., & others. (2023). *Segment Anything*. arXiv preprint arXiv:2304.02643. https://arxiv.org/abs/2304.02643

[32] Gao, X., Li, B., Guo, X., Li, T., Zhang, Y., Li, Y., Wei, Y., Li, X., Li, G., & Wei, X. (2023). *ViTDet: Vision Transformer for Object Detection*. arXiv preprint arXiv:2303.11098. https://arxiv.org/abs/2303.11098

[33] Mage AI. (2021, September 7). *Explorations in Knowledge Distillation.* DEV Community .https://dev.to/mage_ai/explorations-in-knowledge-distillation-3cm3#:~:text=The%20best%20performing%20setting%20by,model%20from%20scratch%2C%20without%20distillation.

[34] Simpson, A. L., Antonelli, M., Bakas, S., Bilello, M., Farahani, K., van Ginneken, B., et al. (2019). *A large annotated medical image dataset for the development and evaluation of segmentation algorithms*. arXiv preprint arXiv:1902.09063.

[35] Wang, Y., Cheng, L., Duan, M., Wang, Y., Feng, Z., & Kong, S. (2023). *Improving Knowledge Distillation via Regularizing Feature Norm and Direction*. arXiv preprint arXiv:2305.17007. https://doi.org/10.48550/arXiv.2305.17007

[36]: Mansourian, A. M., Jalali, A., Ahmadi, R., & Kasaei, S. (2024). *Attention-guided Feature Distillation for Semantic Segmentation*. arXiv preprint arXiv:2403.05451. https://doi.org/10.48550/arXiv.2403.05451

[37] Meng, T., Ghiasi, G., Mahjourian, R., Le, Q. V., & Tan, M. (2022). *Revisiting Multi-Scale Feature Fusion for Semantic Segmentation*. arXiv preprint arXiv:2203.12683. https://doi.org/10.48550/arXiv.2203.12683

[38] Phuong, M., & Lampert, C. H. (2019). Towards understanding knowledge distillation. In K. Chaudhuri & R. Salakhutdinov (Eds.), *Proceedings of the 36th International Conference on Machine Learning* (Vol. 97, pp. 5142–5151). PMLR.

Name of Candidate: Dhruv Varahavenkatasai Rachakonda

Date of Submission: April 16, 2025