

UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

ADVANCEMENTS IN IMAGE SEGMENTATION: FROM SUPERVISED
LEARNING TO ZERO-SHOT FOUNDATION MODELS

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

DOCTOR OF PHILOSOPHY

By

HUONG PHAM
Norman, Oklahoma
2025

ADVANCEMENTS IN IMAGE SEGMENTATION: FROM SUPERVISED
LEARNING TO ZERO-SHOT FOUNDATION MODELS

A DISSERTATION APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Samuel Cheng, Chair

Dr. Choon Yik Tang

Dr. Javier A. Jo

Dr. Tomasz Przebinda

Acknowledgements

I am deeply indebted to my advisor and committee chair, Dr. Samuel Cheng, for his unwavering support, guidance, and encouragement throughout the completion of this dissertation. His insight and mentorship have been instrumental in shaping my research and academic growth, and this work would not have been possible without his invaluable help.

I would also like to express my sincere appreciation to my committee members for their valuable input and constructive feedback, which greatly enhanced the quality of this work, and to Dr. Bin Zheng for his early guidance and valuable discussions that laid the foundation for my research. My appreciation extends to my lab mates — Haoliang Zhang, Reza Babaei, Dong Han, and Zhihao Zhao — whose companionship, collaboration, and thoughtful conversations made my research journey both productive and enjoyable.

I am also grateful to the School of Electrical and Computer Engineering at the University of Oklahoma for its financial and technical support during my Ph.D. study, which greatly contributed to this research.

Finally, I am thankful to my family and loved ones for their patience and understanding, which have been a constant source of motivation throughout this endeavor.

Table of Contents

Abstract	viii
1 Introduction	1
1.1 Background and Motivation	1
1.2 Problem Statements	3
1.3 Aims and Objectives	4
1.4 Contribution	5
1.5 Publications	6
1.6 Organization	7
2 Foundations of Image Segmentation	9
2.1 Conceptual Overview	10
2.2 Motivation and Application Spectrum	10
2.3 Recent Advances in Segmentation Techniques	11
2.3.1 Convolutional Encoder–Decoder Architectures	11
2.3.2 Transformer-Based Segmentation Models	12
2.3.3 Self-Supervised and Semi-Supervised Learning	12
2.3.4 Prompt-Driven and Zero-Shot Segmentation	13
2.3.5 Multi-Modal and Cross-Domain Frameworks	13
2.4 Progressive Development of Segmentation Techniques	14
2.5 Conclusions	15
3 Water Segmentation with Classical Machine Learning Approaches	17
3.1 Introduction	18
3.2 Related Work	20
3.3 Study Site and Datasets	21
3.3.1 Landsat Data	21
3.3.2 Continuous Monitoring of Land Disturbance	23
3.4 Methods	23
3.4.1 Water Bodies Extraction	24
3.4.2 Water and Non-Water Bodies Classifier Using COLD-Derived Data	25
3.4.3 Determining Water Changes at Breakpoints in COLD-derived data	26
3.5 Results	27
3.5.1 Water Frequency Prediction Using a Regression Model	27
3.5.2 Assessing Water Frequency Prediction: Classification vs. Re- gression Models	29
3.5.3 Determine Water Change Using COLD-derived Data	29
3.6 Discussion	32
3.7 Conclusions	34

4	Non-Iterative Cluster Routing	35
4.1	Introduction	36
4.2	Related Work	40
4.2.1	Agreement Routing in Capsule Network	40
4.2.2	Attention Routing	42
4.3	Methods	43
4.3.1	Notation and Terminology	43
4.3.2	Original Non-Iterative Cluster Routing	44
4.3.3	Cluster Routing Implementation	45
4.3.4	Proposed Routing Variations	51
4.4	Experimental Setup	56
4.5	Results	58
4.5.1	Comparison with the Baseline CNN	59
4.5.2	Comparisons with the State-of-the-Art	61
4.5.3	Classification across Various Viewpoints	62
4.5.4	Agreement Indicators	63
4.6	Conclusions	64
5	Enhancing Semantic Segmentation with Reinforced Active Learning	68
5.1	Introduction	69
5.2	Background	70
5.2.1	Semantic Segmentation	70
5.2.2	Active learning	71
5.2.3	Reinforced Active Learning	72
5.2.4	Significance of Reinforced Active Learning in Semantic Segmentation	73
5.3	Related Work	73
5.4	Methodology	75
5.4.1	Active Learning with Reinforcement Learning for Semantic Segmentation	75
5.4.2	Extensions to DQN	77
5.5	Experimental Setup	80
5.5.1	Dataset Description	81
5.5.2	Data Collection and Preprocessing	82
5.5.3	Evaluation Metrics	83
5.5.4	Hardware and Software Configurations	84
5.6	Results	84
5.7	Conclusions	88
6	Prototype-Guided Zero-Shot Medical Image Segmentation with Large Vision-Language Models	90
6.1	Introduction	90
6.2	Background and Related Work	92
6.3	Methodology	94
6.3.1	Overview of the Proposed Framework	94

6.3.2	Bounding Box Estimation with Prompt-Engineered Large Vision-Language Models	96
6.3.3	Integration of LVLM Bounding Boxes with SAM and Prototype-Based Retrieval	97
6.4	Experiments	99
6.4.1	Datasets and Evaluation Metrics	99
6.4.2	Performance	102
6.4.3	Prototype-Based Similarity and LVLM Bounding Box	104
6.4.4	Ablations	105
6.5	Discussion	109
6.6	Conclusions	112
A	Prompt engineering	114
A.1	Brain	114
A.2	Lungs	115
A.3	Fetal heads	117
B	Visual results	119
7	Conclusions	123
	References	126

Abstract

Image segmentation is a fundamental task in computer vision with critical importance in biomedical imaging, environmental monitoring, and artificial intelligence. This dissertation investigates the evolution of segmentation methodologies across four major paradigms: traditional machine learning with handcrafted features, deep learning architectures, reinforcement learning-enhanced active learning, and zero-shot segmentation enabled by large vision-language models. Early work demonstrates that classical pixel-wise classifiers such as random forests and regression models, applied within the COLA framework, can achieve meaningful segmentation and land-change detection despite limited annotated data, though their reliance on handcrafted features restricts scalability. The transition to deep learning, exemplified by U-Net rectum segmentation on low-field MRI, improved accuracy by learning hierarchical features directly from data but remained constrained by small datasets, annotation burden, and limited cross-modality generalization. To reduce labeling costs and address dataset imbalance, this dissertation develops a reinforced active learning framework that integrates reinforcement learning with region-level uncertainty sampling, enabling efficient selection of informative samples and improving segmentation performance under constrained annotation budgets. Finally, the dissertation introduces a prototype-guided zero-shot segmentation framework that integrates bounding box priors from large vision-language models (Gemini Pro 2.5) with mask candidates from SAM and prototype-based similarity matching using CLIP embeddings. This training-free approach achieves robust segmentation across brain MRI, fetal ultra-

sound, and chest X-ray without task-specific fine tuning, reducing the performance gap between zero-shot and supervised models. Collectively, these contributions trace a technological progression from handcrafted feature engineering toward adaptable, annotation-efficient, and training-free segmentation paradigms. The findings highlight the growing potential of foundation models to democratize advanced image analysis in resource-limited environments. Future work includes improving LVLM-generated spatial priors, developing adaptive prototype selection strategies, and extending zero-shot frameworks to broader biomedical and environmental applications.

Chapter 1

Introduction

1.1 Background and Motivation

Image segmentation is a core technique in computer vision and artificial intelligence, involving the division of an image into distinct regions to identify objects, boundaries, or areas of interest. It enables higher-level tasks such as object detection, recognition, and scene understanding. Applications range from autonomous vehicles, which differentiate between roads, pedestrians, and obstacles, to augmented reality, where segmentation integrates virtual objects into real-world environments.

In medical imaging, segmentation isolates anatomical structures, such as organs or tumors, aiding diagnosis, surgical planning, and treatment guidance. For example, rectum segmentation in low-field prostate MRI enhances biopsy precision [1], while segmentation of gastric tumors from CT scans supports radiomic feature extraction and personalized treatment planning [2].

Segmentation is also critical in environmental monitoring and geospatial analysis. Satellite imagery is segmented to map land use, track deforestation, and monitor water bodies, providing insights for urban planning and conservation. In disaster management, segmentation facilitates rapid assessment of affected areas during floods, wildfires, or earthquakes, aiding resource allocation and response efforts.

Challenges in image segmentation arise from the variability and complexity of visual data. Objects often differ in shape, size, and texture, appear under varying lighting

conditions, or overlap, complicating segmentation. Ambiguous boundaries, such as gradual transitions in medical imaging or irregular shapes in natural scenes, require sophisticated models to infer meaningful separations. Contextual understanding, such as recognizing relationships between objects and their surroundings, adds further complexity, especially for real-time applications.

Scale variability poses another challenge, as objects within an image can appear at vastly different sizes. Models must balance capturing fine details with maintaining accuracy for larger structures, increasing computational demands. Additionally, deep learning models, while effective, are computationally intensive, making it difficult to achieve real-time segmentation in resource-constrained settings.

Limited annotated data further complicates segmentation tasks. Producing pixel-level annotations is time-consuming and often requires domain expertise, particularly in fields like medical imaging. This scarcity hinders the development of robust models, especially for specialized applications. Moreover, ensuring model generalization across datasets with varying conditions and equipment remains a persistent issue.

In summary, image segmentation is a complex task requiring robust solutions to handle variability, ambiguity, and computational constraints. Despite these challenges, advancements in segmentation are essential for progress in fields such as healthcare, environmental monitoring, and robotics, driving innovation across diverse domains.

1.2 Problem Statements

Data Creation: One of the most challenging and time-consuming aspects of developing high-performance automated segmentation systems is producing high-quality annotated datasets. For segmentation tasks, this involves generating precise ground truth masks with pixel-level accuracy, a process often performed manually by experts, such as radiologists. While highly accurate, manual annotation is labor-intensive and prone to inconsistencies due to fatigue, particularly for complex boundaries. Semi-automatic methods, such as using pretrained models to generate initial masks for correction, reduce labor but still require significant user input. In addition, real-world datasets often exhibit class imbalances, which bias models toward dominant classes and reduce generalizability. To address these issues, active learning focuses on annotating the most informative samples. By targeting small, ambiguous, or hard-to-classify image regions instead of entire images, it minimizes redundant annotations and enhances decision boundary learning. This approach improves model generalization, mitigates class bias, and optimizes expert time by prioritizing impactful samples, ensuring more efficient and effective data creation.

Training Issues: Training deep learning models can often be inefficient due to the lack of sufficient data to capture the complexity and variability of the problem domain. This limitation frequently leads to overfitting, where the model learns patterns and noise specific to the training data, resulting in poor generalization to unseen data. Additionally, medical image datasets are highly diverse, varying across imaging modalities (e.g., MRI, X-ray, ultrasound) and anatomical focuses (e.g., brain, lungs,

fetal head). Training separate models for each dataset or segmentation task exacerbates inefficiency and resource demands. A training-free solution leveraging zero-shot methods addresses these challenges by reducing the need for extensive data collection and training. Such an approach is scalable and adaptable to new segmentation tasks without requiring retraining, which is crucial for rapidly evolving medical applications. This adaptability is made possible by the generalization capabilities of large vision-language models like SAM, trained on a vast dataset of over a billion masks. SAM can handle a wide range of image segmentation tasks without the need for task-specific training, making it a powerful tool for diverse and complex segmentation challenges.

1.3 Aims and Objectives

The aim of this research is to develop efficient methodologies for image segmentation, focusing on automation and optimizing performance under constrained resources. The objectives are to:

1. design and develop novel neural architectures that enhance performance and can be applied to segmentation tasks, ensuring scalability and robustness.
2. address challenges of limited annotated datasets by using active learning to prioritize informative data and mitigate class imbalance for improved generalization;
3. reduce training workload and annotation requirements through a zero-shot approach while maintaining or improving segmentation performance.

1.4 Contribution

This dissertation contributes to the field of image segmentation in the following key areas:

- Non-Iterative Cluster Routing for Capsule Networks:
 - Enhanced capsule-based architectures with routing variants that eliminate iterative overhead, achieving state-of-the-art performance.
 - Extended the approach to 3D U-Net segmentation.
- Enhanced Reinforcement Learning-Based Active Learning Framework:
 - Improved an active learning framework to prioritize labeling the most informative and underrepresented regions.
 - Directly addressed class imbalance issues, enhancing segmentation performance with fewer annotations.
 - Reduced the high cost and time requirements associated with annotation.
- Zero-Shot Segmentation with Foundational Vision Models and LLMs:
 - Introduced a novel approach integrating foundational vision models (e.g., SAM, CLIP) and large language models (LLMs).
 - Achieved alignment with ground truth without requiring additional annotations or extensive training.

These contributions advance the field of segmentation by improving performance, reducing dependency on large annotated datasets, and offering scalable, efficient solutions for complex biomedical and environmental applications.

1.5 Publications

The research in this thesis is supported by the following published papers:

- H. Pham and S. Cheng. Prototype-Guided Zero-Shot Medical Image Segmentation with Large Vision-Language Models, *Applied Sciences Journal*, 15(21):11441, 2025.
- H. Pham, S. Cheng, T. Hu, and C. Deng. Continuous Monitoring of Waterbody Change Detection in Oklahoma Using Landsat Time Series, *Remote Sensing Letters Journal*, 16(11):1216–1226, 2025.
- H. Pham, S. Cheng. Non-Iterative Cluster Routing: Analysis and Implementation Strategies, *Applied Sciences Journal*, 14 (5), 1706 (2024).
- Han D, H. Pham, Samuel C. Enhancing Semantic Segmentation through Reinforced Active Learning: Combatting Dataset Imbalances and Bolstering Annotation Efficiency, *Journal of Electronic and Information Systems*, 2023, 5(2): 45-60.
- H. Pham, D. B. T. Le, M. Sadinski, R. Narayanan, A. Nacev, and B. Zheng. Deep Learning Based Rectum Segmentation on Low-field Prostate MRI to Assist

Image-guided Biopsy. In *Medical Imaging 2023: Image-Guided Procedures, Robotic Interventions, and Modeling, SPIE*.

- H. Pham, M. Jones, T. Gai, W. Islam, G. Danala, J. Jo, and B. Zheng. Identifying an optimal machine learning generated image marker to predict survival of gastric cancer patients. In *Medical Imaging 2022: Computer-Aided Diagnosis, SPIE*.

1.6 Organization

The rest of the dissertation is organized as follows:

Chapter 2 provides the fundamental concepts of image segmentation, including traditional methods and recent advances, to establish the foundation for the subsequent chapters.

Chapter 3 explores novel environmental application, including water body mapping and change detection using the Continuous Monitoring of Land Disturbance (COLD) algorithm.

Chapter 4 improves the cluster routing algorithm for capsule-based classification tasks and demonstrates its applicability to 3D image segmentation.

Chapter 5 presents reinforced active learning strategies to improve semantic segmentation, tackling challenges like dataset imbalance and annotation efficiency.

Chapter 6 presents a novel zero-shot segmentation framework that integrates foundational segmentation models with language-vision models to improve segmentation accuracy, reduce annotation costs, and accelerate product development.

Finally, Chapter 7 summarizes the key findings, highlights the contributions, and

discusses implications and future research directions, demonstrating the impact of this dissertation across biomedical and environmental domains.

Chapter 2

Foundations of Image Segmentation

Before exploring specific applications, it is essential to position image segmentation within its broader historical and methodological progression. Early segmentation techniques relied on heuristic rules such as thresholding, region growing, and watershed algorithms, which offered simplicity but lacked robustness under real-world variability. The field then shifted toward machine learning with handcrafted features—spectral indices, texture descriptors, and edge detectors—combined with classifiers like support vector machines, decision trees, and random forests to perform pixel-wise labeling. While these approaches enabled scalable analysis in both environmental monitoring and medical imaging, they were limited by the need for extensive feature engineering and their sensitivity to noise and domain shifts. The advent of deep learning, particularly convolutional encoder-decoders such as U-Net, marked a turning point by allowing models to learn hierarchical features directly from data, greatly improving accuracy and generalization. More recently, active learning strategies have been introduced to reduce annotation costs, and zero-shot frameworks leveraging foundation models have opened the door to training-free segmentation across diverse modalities. This trajectory underscores the importance of understanding the theoretical underpinnings of segmentation, which remain central to both classical and modern approaches. The following sections provide a structured overview of the fundamentals, motivations, methodologies, and recent advancements in segmentation, establishing the conceptual foundation for the chapters that follow.

2.1 Conceptual Overview

Image segmentation refers to the systematic partitioning of an image into spatially coherent and semantically meaningful regions, each representing distinct entities such as anatomical structures, land-cover classes, or physical objects. In contrast to image classification, which determines what is present within an image, and object detection, which identifies where objects are located, segmentation delineates the precise pixel boundaries that define which pixels belong to each entity. This pixel-level granularity enables computational models to achieve a level of perceptual understanding analogous to that of human vision.

Conceptually, segmentation can be viewed as a process of enforcing semantic consistency across pixels that share similar contextual or structural attributes. By assigning uniform labels to all pixels within a given region, the algorithm converts raw visual data into structured, interpretable representations that facilitate downstream tasks such as quantitative analysis, pattern recognition, and automated decision-making.

2.2 Motivation and Application Spectrum

The motivation for image segmentation arises from the need to transform raw visual data into structured information that supports analysis, interpretation, and decision-making across a broad range of domains. In medical imaging, segmentation enables the delineation of anatomical structures such as organs, lesions, and tumors, which is essential for diagnosis, surgical planning, and treatment monitoring. In environmental

monitoring, it facilitates the classification of land cover, the detection of deforestation, and the assessment of water dynamics, thereby contributing to sustainable resource management and climate studies. Within robotics and augmented or virtual reality, segmentation provides the spatial understanding necessary for autonomous navigation, object manipulation, and realistic human–environment interaction.

Despite its broad applicability, segmentation remains a challenging problem due to the inherent variability and complexity of visual data. Objects may differ in shape, illumination, texture, and scale, while boundaries often exhibit ambiguity or occlusion. These factors demand the development of models capable of learning contextual relationships, adapting to diverse domains, and generalizing beyond the examples encountered during training.

2.3 Recent Advances in Segmentation Techniques

Recent advances in image segmentation have been driven by the convergence of deep learning architectures, attention mechanisms, and large-scale pretraining. These innovations have transformed segmentation from task-specific modeling into a foundation-driven paradigm capable of generalizing across domains and modalities. This section outlines several of the most prominent segmentation techniques that have shaped the contemporary landscape.

2.3.1 Convolutional Encoder–Decoder Architectures

Convolutional encoder–decoder frameworks remain the backbone of modern segmentation. Among them, the U-Net architecture and its variants—such as U-Net++ [3],

Attention U-Net [4], and ResU-Net [5]—have demonstrated exceptional performance in medical and environmental imaging. These models employ symmetric contracting and expanding paths to capture both global context and fine spatial detail [6]. The skip connections between encoder and decoder layers preserve boundary information and mitigate the loss of resolution typically associated with downsampling operations. Extensions of U-Net have further integrated residual learning, attention gating, and multi-scale fusion to improve robustness under noise and variability [7].

2.3.2 Transformer-Based Segmentation Models

The introduction of the Vision Transformer (ViT) [8] and its derivatives has significantly influenced segmentation research. Transformer-based architectures such as SegFormer [9], Swin-Unet [10], and Mask2Former [11] leverage self-attention mechanisms to capture long-range dependencies between pixels, surpassing the locality constraints of convolutional networks. These models exhibit strong generalization capabilities and have become central to large-scale semantic and instance segmentation benchmarks. The hybridization of CNN backbones with transformer modules has also emerged as a practical compromise between computational efficiency and representational power.

2.3.3 Self-Supervised and Semi-Supervised Learning

Given the cost and difficulty of obtaining pixel-level annotations, self-supervised and semi-supervised learning approaches have gained increasing attention. Self-supervised pretraining leverages auxiliary tasks—such as image reconstruction, contrastive rep-

resentation learning, or masked image modeling—to learn rich feature embeddings without manual labels [12, 13, 14]. Semi-supervised methods extend this by incorporating limited labeled data alongside large unlabeled datasets, employing techniques such as consistency regularization, pseudo-labeling, and entropy minimization to guide the learning process [15, 16]. These strategies have proven particularly effective in medical imaging and remote sensing, where labeled data is scarce.

2.3.4 Prompt-Driven and Zero-Shot Segmentation

A major paradigm shift has emerged with the advent of foundation models and zero-shot segmentation. Models such as the Segment Anything Model (SAM) [17] and CLIPSeg [18] leverage large-scale pretraining on diverse datasets to perform segmentation without task-specific fine-tuning. SAM utilizes a prompt-based mechanism—accepting text, points, or bounding boxes—to generate high-quality masks adaptable to arbitrary objects and scenes. CLIPSeg and related vision–language models extend this concept by aligning textual and visual embeddings, enabling segmentation guided by semantic prompts. These approaches redefine segmentation as a general-purpose reasoning task rather than a fixed classification problem.

2.3.5 Multi-Modal and Cross-Domain Frameworks

Recent research trends also emphasize multi-modal and cross-domain segmentation, integrating data from complementary sources such as MRI–CT pairs, RGB–depth images, or optical–SAR satellite observations. Models such as MedSegDiff [19] and UNETR [20] exploit multimodal fusion and transformer-based encoders to enhance

robustness against domain shifts. Furthermore, domain adaptation techniques and adversarial training have been applied to transfer knowledge across datasets with differing visual characteristics, reducing dependence on labeled data in each target domain [21, 22, 23, 24].

2.4 Progressive Development of Segmentation Techniques

The evolution of image segmentation reflects a progressive shift from handcrafted heuristics toward data-driven and generalizable frameworks. Early segmentation methods were largely heuristic and rule-based, relying on techniques such as thresholding, edge detection, region growing, and watershed algorithms. While these approaches were computationally efficient and conceptually transparent, they exhibited limited robustness under variations in illumination, noise, and object morphology.

The next methodological phase introduced feature engineering and classical machine learning, where segmentation was formulated as a pixel-wise classification problem. Handcrafted descriptors—such as spectral indices, texture measures, and edge-based features—were combined with classifiers including support vector machines, decision trees, and random forests. Although these methods enhanced flexibility and scalability, their reliance on domain-specific feature design and extensive parameter tuning constrained their generalization across datasets and modalities.

A major paradigm shift occurred with the advent of deep learning, particularly convolutional encoder-decoder architectures such as U-Net. These models learned hierarchical feature representations directly from data, capturing both local spatial

details and global contextual relationships. This data-driven learning process led to dramatic performance improvements in domains ranging from medical imaging to environmental monitoring.

Subsequent developments incorporated active and reinforcement learning strategies to address the challenge of annotation scarcity. By identifying the most informative samples for labeling, these methods reduced the cost and effort associated with data curation while maintaining model accuracy.

Most recently, the emergence of zero-shot and foundation model approaches has marked a new era of segmentation research. Foundation vision and vision–language models leverage large-scale pretraining to perform segmentation tasks without explicit task-specific supervision. These prompt-driven systems exemplify a transition toward general-purpose, adaptable frameworks capable of understanding and segmenting diverse visual scenes with minimal fine-tuning.

This methodological trajectory underscores the field’s movement from rigid, rule-based techniques toward adaptive, semantically aware systems that integrate context, learning, and reasoning within a unified framework.

2.5 Conclusions

This dissertation presents an integrated progression of research that bridges classical and modern segmentation methodologies across biomedical and environmental domains. It spans four interconnected directions: (1) pixel-wise classification and segmentation using classical machine learning for environmental monitoring with

time-series analysis, (2) deep learning approaches with capsule networks and U-Net architectures for medical image segmentation, (3) reinforced active learning to minimize annotation cost and improve data efficiency, and (4) zero-shot segmentation with large vision-language models to achieve annotation-free and robust performance. Collectively, these studies address key challenges of data scarcity, annotation burden, domain variability, and model generalization, contributing to more scalable and adaptive segmentation frameworks.

Having reviewed these advances, it is important to recognize that modern deep learning and foundation models grew out of earlier traditions of segmentation research. Before convolutional networks and large vision language models dominated the field, segmentation was typically performed through pixel wise classification using handcrafted features such as spectral indices, edge detectors, and texture descriptors. Classical machine learning algorithms including decision trees, random forests, and support vector machines were then applied to assign class labels at the pixel level. These methods represented a natural evolution from heuristic techniques such as thresholding or region growing and provided the first scalable frameworks for environmental and biomedical applications. Building on this historical foundation, the next chapter turns to one such application: pixel wise water frequency estimation using COLD derived features and classical machine learning, illustrating both the strengths and the limitations of handcrafted feature based segmentation in comparison to more recent approaches.

Chapter 3

Water Segmentation with Classical Machine Learning Approaches

Building on the theoretical foundations and related works on image segmentation discussed in Chapter 2, this chapter turns to one of the earliest traditions in segmentation research: pixel-wise classification using handcrafted features and classical machine learning. Before the advent of deep neural networks and foundation models, segmentation was typically framed as a supervised classification task, where each pixel was described by manually designed features such as spectral indices, texture descriptors, or edge statistics, and then assigned a class label using algorithms like decision trees, random forests, or support vector machines. These methods established the baseline for many environmental and biomedical applications, providing an important bridge between simple thresholding or region growing techniques and today’s advanced deep learning architectures.

In this context, we focus on environmental monitoring and specifically water mapping, applying classical regression and classification models to features derived from the Continuous Monitoring of Land Disturbance (COLD) algorithm. This study highlights both the historical importance and the continuing relevance of handcrafted-feature-based methods: while less flexible than modern deep learning approaches, they offer interpretability, computational efficiency, and robustness in settings where annotated training data is scarce. The following sections introduce the datasets, methodology, and results of this work.

3.1 Introduction

The Continuous Monitoring of Land Disturbance (COLD) algorithm, developed using Landsat time series data, effectively detects land disturbances as new imagery becomes available and supports historical disturbance mapping [25]. Refinements such as using surface reflectance instead of Top of Atmosphere, integrating multiple spectral bands, and removing outliers based on model predictions have improved detection performance. The algorithm also distinguishes true disturbances from vegetation regrowth, enhancing mapping accuracy. These capabilities provide a solid basis for estimating water frequency, defined as the proportion of observations classified as water among stable periods, enabling pixel-level delineation of water bodies and analysis of water trends.

Generating land cover classifications from remote sensing data is a relatively straightforward process, but achieving high accuracy presents challenges. To improve classification accuracy, many products incorporate multi-temporal images into their algorithms [26]. However, automating the collection of multi-temporal images can introduce several potential issues. One major challenge is ensuring data consistency across images. Variations in atmospheric conditions during different image acquisition dates can affect spectral signatures, making it difficult for algorithms to maintain consistent classification over time. For example, cloud-free images are essential for accurately classifying each pixel across multiple observations. This requirement is often unattainable, particularly for sensors with low temporal frequency, such as Landsat. As a result, acquiring suitable images can take several years, leading to the

production of Landsat-based land cover maps at five- or ten-year intervals [27].

Additionally, temporal variability in images, such as changes in vegetation or land use due to human activities, can lead to misclassification if the algorithm does not account for these temporal dynamics [28]. This issue is particularly pronounced when images from widely spaced intervals are used or when mapping areas that undergo frequent changes. Consequently, land cover maps produced using conventional multi-temporal methods may be unreliable for identifying land cover changes, as classification errors can be misinterpreted as actual changes. This problem is further exacerbated in smaller areas where subtle changes are more likely to occur.

To address these challenges, using data derived from Continuous Monitoring of Land Disturbance (COLD) algorithm could be a promising approach for classifying water bodies, as it (1) accounts for the temporal dynamics of land cover changes, (2) is robust against data inconsistencies, and (3) reliably detects water change trends. The main contributions of this study are as follows:

1. This is the first study to adapt the COLD algorithm, originally developed for detecting general land disturbances, for the specific purpose of monitoring water frequency and change at the pixel level.
2. We propose a novel methodology that applies well-established classification and regression models to newly constructed features derived from COLD algorithm, enabling effective estimation of water frequency and long-term trend detection.
3. We introduce a framework to estimate changes in water frequency by comparing conditions before and after breakpoints identified in the COLD algorithm,

enabling the detection of water increases, decreases, or stable trends.

3.2 Related Work

Various methods for extracting water surfaces using remote sensing data have been developed, including both machine learning and traditional approaches [29]. Machine learning methods require large datasets and careful sample selection to ensure generalization across different regions. Traditional algorithms, in contrast, often rely on spectral differences between water and non-water areas, using multiple bands for effective extraction. A notable example is the water index method, which combines spectral bands and remote sensing indices to distinguish water bodies. This method has demonstrated high performance in studies and offers advantages such as simplicity, robustness, and ease of deployment across large, diverse areas, enabling quick and efficient extraction [30].

Deep learning, particularly with convolutional neural networks (CNNs), has proven useful for segmentation tasks in remote sensing image analysis. For example, the CNN-based model U-Net has been widely used in segmentation tasks [31]. While deep learning models are powerful for tasks such as water body extraction, they are heavily data-dependent and require large amounts of labeled data for training. Acquiring large, continuous, high-quality labeled datasets can be challenging and time-consuming [29]. Due to the difficulty of acquiring high-quality training datasets, these models can be prone to overfitting, especially when the dataset is small or not representative of the broader context, leading to poor generalization on unseen data [32].

These models also face challenges, such as the need for cloud-free, high-resolution images, which are difficult to obtain consistently for long-term water change detection [33, 31]. Many methods rely on individual images for segmentation models, making them sensitive to cloud cover. However, the Continuous Change Detection and Classification (CCDC) model by [27], which uses all Landsat data, addresses this by building a dynamic time series model that detects various land cover types, including water. Its updated version, Continuous Monitoring of Land Disturbance (COLD), enhances small-change detection, automates processing, ensures model stability, and handles cloud and snow effects [25]. Our proposed model leverages this data to estimate water indices, classify water bodies, and monitor water trends across regions.

3.3 Study Site and Datasets

3.3.1 Landsat Data

Monitoring surface water is crucial for understanding ecological and hydrological processes. Recent advances in satellite-based optical remote sensing have significantly improved surface water detection [34]. A common method involves using Landsat remote sensing data and water indices to track changes in water bodies throughout the year over large areas [33, 35].

This study uses Landsat-5 TM imagery (1984–2013), which offers the longest continuous record among Landsat missions, with consistent sensor characteristics ideal for long-term monitoring. Landsat was chosen for its long-term availability, moderate resolution, and suitability for our spatial and computational needs. Cloud

cover is a major challenge in using Landsat imagery. Incomplete masking of clouds or shadows can lead to misclassifications, introducing false positives or negatives in water body segmentation. Our method is inherently robust to occasional gaps in monthly observations due to cloud contamination. Since water frequency is estimated by averaging valid water index values across available months, the absence of imagery in one or more months has a minimal impact on overall accuracy. This temporal averaging approach, combined with COLD’s stable-period modeling, allows for reliable classification even when some observations are missing, reducing the sensitivity to intermittent cloud cover.

Fig. 3.1: Geographic coordinates and counties of the study regions within Landsat tile h12v16

Lake Name	Latitude (°N)	Longitude (°W)	County
Lake Stanley Draper	35.3413	97.3519	Cleveland
Shawnee Reservoir	35.3388	97.0789	Pottawatomie
Lake Carl Blackwell	36.1340	97.2240	Payne
Lake McMurtry	36.1759	97.1796	Noble
Lake Perry	36.2677	97.2760	Noble
Lake Langston	35.9244	97.2723	Logan
Liberty Lake	35.7950	97.4615	Logan
Guthrie Lake	35.8131	97.4379	Logan

3.3.2 Continuous Monitoring of Land Disturbance

The COLD data aligns with the tiling system used by the Landsat Analysis Ready Data (ARD) [36] across the Continental United States (CONUS), which consists of 427 mapped tiles. Each tile references the Albers Equal Area Conic projection and is based on the WGS84 datum. The tiles are uniformly sized at 5,000 by 5,000 pixels, with each pixel representing a 30-meter square, covering an area of 150 by 150 kilometers per tile. Tile identification is facilitated through horizontal ('h') and vertical ('v') coordinates.

Our specific focus is on tile h12v16, located in Oklahoma, which was selected for its availability of COLD data. The COLD data is aggregated into a 5000x5000 matrix file. Each pixel contains one or more stable time points based on multiple coefficients.

3.4 Methods

This section outlines the methodology for building a dataset to train machine learning models for characterizing water surfaces from 1984 to 2013. Water body maps were generated and combined with COLD data to develop models that estimate average water index or frequency. This approach enables clear separation of water and non-water bodies and captures pixel level changes in water trends. The overall workflow is shown in Figure 3.1.

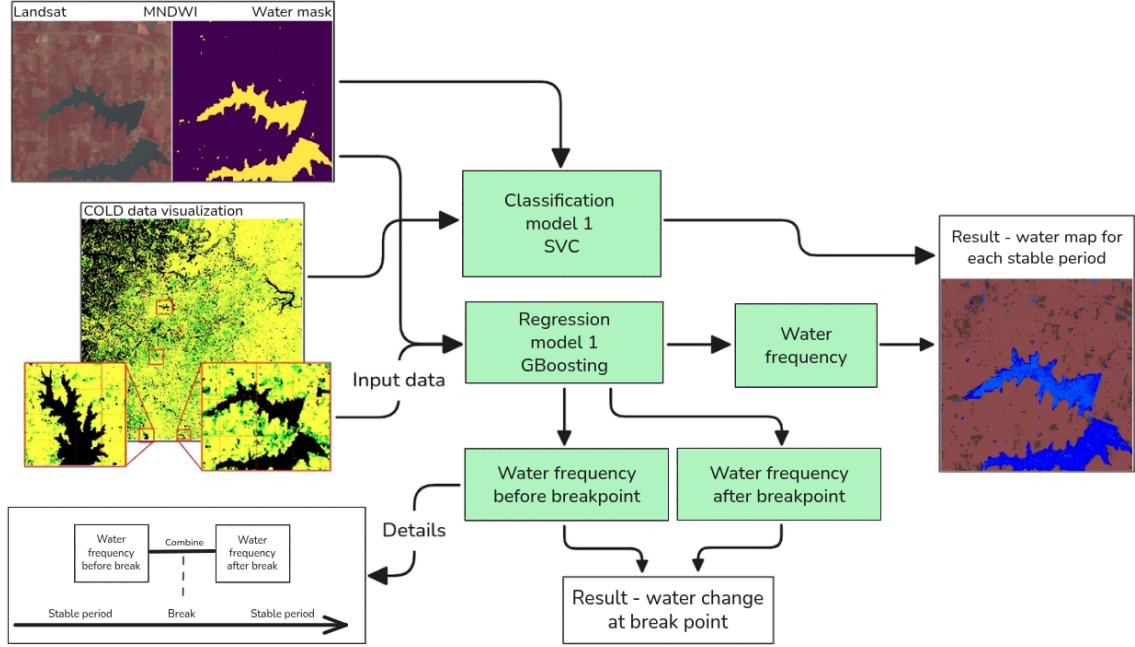


Fig. 3.1: Methodology to predict the water mapping and water change from COLD data.

3.4.1 Water Bodies Extraction

Water indices, derived from two or more spectral bands, provide a straightforward and effective method for water extraction [33, 34]. Various indices can estimate surface water areas or flood sizes. The MNDWI is effective for this task using the shortwave infrared band (SWIR) for better detection of minor water features and being less affected by sediment and other water constituents, making it more reliable [37]. These indices have proven effective in multiple studies [38], providing a solid foundation for building the input data for training models to utilize COLD data for detecting water surfaces and their changes over time.

To build the dataset for training machine learning models to characterize water surfaces from 1984 to 2013, the MNDWI was used to extract water features. It was

widely adopted for its efficiency and reliability, enabled the extraction of water and land masks from lake images. A manual threshold of zero, chosen for its simplicity and accuracy, classified pixels as water or non-water. This threshold served as the ground truth to assess the correlation between COLD data and water bodies. Using Landsat 5 data, the MNDWI was calculated from bands 2 and 5 [39].

3.4.2 Water and Non-Water Bodies Classifier Using COLD-Derived Data

Water indices extracted monthly from stable periods of COLD derived data are used to compute water frequency or average indices over multiple years. Each index reflects the MNDWI value for a pixel during a stable period. Water frequency is defined as the proportion of months a pixel is classified as water, ranging from 0 (no water observed) to 1 (consistently water). It is calculated by averaging available monthly water observations within each year.

Each pixel has 56 coefficients corresponding to different stable periods from the COLD-derived data, which are used as input features to estimate water frequency within these intervals. To mitigate potential errors arising from uncertainties in image processing, pixels with a water frequency of less than 0.25 are classified as non-water (land) areas, while pixels with a water frequency greater than 0.25 are categorized as water bodies [35, 33].

Leveraging the water index as a benchmark and employing COLD-derived data as input features, we developed regression and classification models to estimate water frequency and distinguish between water and non-water bodies for each pixel. These models include gradient boosting for the regression model and Support Vector Machine

for the classification model. Based on the outcomes, we can determine whether COLD-derived data provides sufficient information to estimate water frequency and accurately classify each pixel as either water or non-water.

3.4.3 Determining Water Changes at Breakpoints in COLD-derived data

COLD-derived pixel data captures multiple coefficients for several stable periods. Breakpoints in the COLD-derived data are identified using the detection logic from the Continuous Change Detection and Classification (CCDC) algorithm, on which COLD is based. Specifically, a change is flagged when the average of the normalized residuals—computed as the difference between observed values and model predictions, divided by three times the RMSE—for all seven Landsat bands exceeds a threshold of 1 for three consecutive clear observations[27]. A breakpoint marks the transition between periods where the previous model no longer fits the new Landsat time series, requiring the model’s coefficients to change to reflect spectral variations, as shown in Figure 3.1. In this experiment, we input the COLD coefficients of pixels with breakpoints into a regression model to predict water frequency before and after the break. This analysis helped us identify whether the change was due to shifts in the water frequency across the stable periods surrounding the breakpoint, as illustrated in Figure 3.1.

The three classes used in this experiment were determined by the absolute difference in the predicted water frequency index between periods before and after the break. To minimize computational error, the first class, labeled as 0, was assigned to cases where the absolute difference was close to zero or less than 0.25. Pixels with a difference

greater than 0.25 were classified as an increasing water frequency class, whereas those with a difference smaller than -0.25 were classified as a decreasing water frequency class.

3.5 Results

This section evaluates classification and regression models for estimating water frequency and segmenting water bodies using COLD-derived data, with ground truth from Landsat-based water indices. Models are trained on six regions (Table 3.1) and tested on a seventh using 7-fold cross-validation to ensure robustness.

3.5.1 Water Frequency Prediction Using a Regression Model

In this experiment, the input to our models consists of water frequency data across available years during the stable period of the COLD-derived dataset. We apply a gradient boosting regression model to estimate water frequency for each pixel during these stable periods, facilitating the identification of whether pixels represent water bodies or non-water bodies. For inference, data from one region is held out while the model is trained on the remaining six regions of interest.

The normalized mean squared error (NMSE) of our regression model is reported with an average value of 0.43 and a 95% confidence interval of 0.12 to 0.75. This indicates that the model’s predictive performance surpasses that of a baseline model, which would simply predict the mean of the actual values. The model demonstrates strong predictive power, as it captures a significant portion of the variance in the data. However, the upper bound of 0.75 suggests that the model exhibits considerably

higher error in some cases, likely due to variations in environmental conditions across different regions. We illustrate the model’s ability to segment water bodies by applying a 0.25 threshold to the predicted water frequency. Pixels below this threshold are classified as non-water, and those above are classified as water bodies [35].

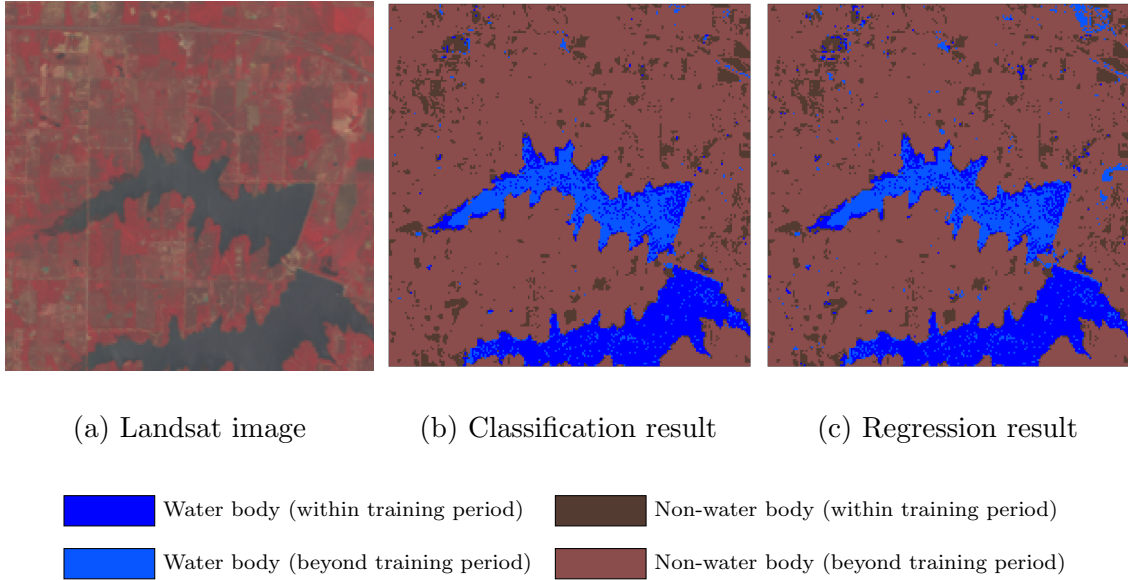


Fig. 3.2: Visualization of water maps generated by the classification (middle) and regression (right) models, compared to the Landsat image (left) as reference. The color scheme differentiates between water and non-water predictions, and indicates whether each pixel’s observation falls within the temporal training period (1984–2001) or in the future inference period (2002–2013). While not explicitly marked in this figure, the training and testing also occur in different geographic regions to evaluate both temporal and spatial generalization

3.5.2 Assessing Water Frequency Prediction: Classification vs. Regression Models

In this experiment, we compared classification model with the regression model to delineate water frequency. For ground truth, we used a threshold of 0.25 to classify pixels as either water bodies or non-water bodies [35]. Among the various models, we selected Support Vector Classification (SVC) as the best performer for classifying water bodies. Our classification result comes out as shown in Table 3.2. The model performs reliably for clear cases, such as low and high-frequency water bodies.

Additionally, our results indicate that the classification model, specifically SVC, achieves marginally higher performance than the best regression model (Gradient Boosting), with an overall accuracy of 0.92 ± 0.06 versus 0.91 ± 0.03 , and an F1 score for the minority class (pixels with water frequency, WF, greater than 0.25) of 0.79 ± 0.11 versus 0.72 ± 0.09 , showing a slightly more balanced performance in Table 3.2. However, the confidence intervals overlap, indicating no statistically significant difference. The visualization between regression and classification map inference is shown in Figure 3.2. This comparison reveals that the regression model tends to over segment the water bodies, as evidenced by the incorrect segmentation in the top right region of the regression map.

3.5.3 Determine Water Change Using COLD-derived Data

This section analyzes pixel data with breakpoints between stable periods to determine whether water frequency trends increase or decrease. Pixels with breakpoints are filtered and used in a regression model to predict water frequency based on COLD

Fig. 3.2: Cross-validation water-body classification metrics across regression and classification models.

	Regression Models			Classification Models		
	GBoosting	RForest	LRegression	XGBoost	RForest	SVC
Overall Acc.	0.91±0.03	0.90±0.04	0.89±0.07	0.92±0.06	0.89±0.06	0.92±0.06
WF ≤ 0.25	0.91±0.04	0.89±0.06	0.85±0.12	0.96±0.04	0.88±0.10	0.94±0.06
WF > 0.25	0.81±0.07	0.85±0.07	0.92±0.07	0.74±0.17	0.81±0.13	0.81±0.07
F1-score	0.72±0.09	0.72±0.11	0.70±0.19	0.77±0.15	0.70±0.21	0.79±0.11

Note. Bold values indicate the best-performing models within each category (regression or classification), based on either the highest overall accuracy or the highest F1-score for the minority class (WF > 0.25). WF = Water Frequency; GBoosting = Gradient Boosting; RForest = Random Forest; LRegression = Linear Regression; XGBoost = Extreme Gradient Boosting; SVC = Support Vector Classification.

coefficients before and after the breakpoint. Predicted frequency changes are classified as a decrease (less than -0.25), an increase (greater than 0.25), or unchanged (between -0.25 and 0.25). This method helps interpret significant changes at the breakpoint and how COLD coefficients respond. By analyzing these coefficients, we aim to predict the water frequency trend for each pixel at the breakpoint. In addition, we also do experiments for determining water change using classification models instead of regression models. In this, we concatenate all COLD coefficients of before and after breakpoint together and use them to predict the differences between known water frequency before and after the breakpoint with the same range previously defined.

A comparison of regression and classification models shows that the regression model consistently outperforms the classification model across all metrics. The regression model’s overall accuracy is 0.81 ± 0.09 , compared to 0.77 ± 0.14 for the classification model, with a narrower confidence interval, indicating greater stability. In the moderate water frequency range ($-0.25 \leq WF \leq 0.25$), the regression model achieves 0.85 ± 0.09 , outperforming the classification model’s 0.82 ± 0.14 . For rare water frequency classes ($WF < -0.25$ and $WF > 0.25$), the regression model again performs better, with accuracies of 0.50 ± 0.11 and 0.42 ± 0.14 , compared to 0.43 ± 0.15 and 0.31 ± 0.12 for the classification model. While both models struggle with these extreme cases, the regression model consistently shows stronger predictive power. It encounters challenges primarily when water frequency significantly changes, as these events are relatively rare.

3.6 Discussion

This study demonstrates the effectiveness of regression and classification models in estimating water frequency and segmenting water bodies using COLD derived data [25]. The gradient boosting regression model achieved a normalized mean squared error of 0.43, with higher errors in some regions likely due to environmental variability. Using a 0.25 threshold, it reached 90% accuracy in segmentation, performing better in low water frequency areas (92%) than in high frequency regions (77%). The classification model slightly outperformed regression (91% vs. 90%), though not significantly. The regression model tended to over segment in some areas. Despite limited training data, both models predicted water frequency beyond the training period (Figure 3.2). COLD coefficients correlated with water presence and detected land disturbances, offering insights into climate change impacts on water bodies.

The regression model outperformed the classification model in detecting water frequency changes at breakpoints, with an accuracy of 81% versus 77%. It was more effective at capturing moderate changes but struggled with extreme variations, likely due to imbalanced training data and regional differences in data distribution. Despite these challenges, it consistently identified water frequency trends more accurately. Future work should consider augmenting the dataset or generating realistic synthetic samples to improve detection of significant changes [40].

Our results are consistent with previous studies on land cover classification, such as [27], which employed the Continuous Change Detection and Classification (CCDC) algorithm—an earlier version of COLD—to classify land cover types using a clas-

sification model. In this study, we utilized a regression model to estimate water frequency, comparing it with a classification model to map water bodies and analyze trends based on data from the updated CCDC algorithm (COLD). Direct comparisons between our approach and previous work using the CCDC model are not feasible due to differences in data sources (CCDC-derived vs. COLD-derived data) and variations in landscape and data collection methods. The primary contribution of our research is the validation and extension of COLD’s capacity to detect water bodies and identify regional trends, which supports water resource management.

To evaluate the performance and robustness of our method, we compared it with two widely used water extraction techniques: the Normalized Difference Water Index (NDWI) and the Modified Normalized Difference Water Index (MNDWI). Using a representative year and a 0.25 threshold on the average annual index, we assessed classification accuracy using the intersection over union (IOU) metric. Our method achieved an IoU of 0.913 ± 0.044 , comparable to MNDWI (0.903 ± 0.023) and higher than NDWI (0.881 ± 0.041). While the accuracy gain is modest, our approach offers practical advantages by leveraging time series coefficients from stable periods, eliminating the need for monthly cloud-free composites. Compared to deep learning models that require large labeled datasets and frequent retraining, our model is lightweight, scalable, and data efficient. Generalizability was demonstrated through cross-validation across seven regions, with consistent IoU values and narrow confidence intervals, particularly for moderate water frequency changes, highlighting its resilience to environmental variability. The model trained on 1984 to 2001 data also generalized well to 2002 to 2013, supporting its utility for long term water monitoring.

3.7 Conclusions

This study demonstrates the effectiveness of the COLD algorithm for water frequency estimation and water body segmentation in aquatic ecosystems. Both regression and classification models applied to coefficients of COLD-derived data produce reliable results, with the regression model achieving a normalized mean squared error of 0.43 and 90% accuracy in water mapping. Despite limited training data, our model accurately predicts water frequency beyond the training period. These results align with prior studies on land cover classification using the CCDC algorithm, extending COLD’s applicability to diverse regions. The regression model also detects water frequency changes at breakpoints with 81% accuracy, though it faces challenges with extreme changes due to imbalanced data. Future research could improve accuracy by incorporating higher-resolution indicators and natural imagery, strengthening its use in environmental monitoring and conservation.

More broadly, this chapter marks an early stage in the evolution of image segmentation, defined by pixelwise classification using handcrafted features and classical machine learning. These methods offered interpretability and computational efficiency but lacked the adaptability needed to handle complex visual variability. The next chapter builds on this foundation by introducing a deep learning-based segmentation framework that incorporates a neural network module called cluster routing. Initially developed for classification tasks, this component now functions as a core element within the U-Net architecture, enabling multiscale contextual learning and representing a major advancement in biomedical imaging.

Chapter 4

Non-Iterative Cluster Routing

Segmentation once leaned on handcrafted cues and familiar classifiers, where pixel-wise labeling and feature engineering formed the foundation of early visual analysis. Following the classical pixelwise classification approaches presented in Chapter 3, this chapter moves into the next stage of segmentation research: deep learning architectures. Unlike handcrafted feature based methods, deep learning models learn hierarchical representations directly from data and are capable of capturing both local detail and global context. Among these, the U-Net has emerged as one of the most influential architectures for medical image segmentation, owing to its encoder–decoder structure and its capacity to integrate fine grained boundary information with semantic understanding.

This chapter steps into a different grammar of vision, where capsules carry pose and relational information, and routing encodes agreement among features. Through non iterative cluster routing and two streamlined variations, the discussion seeks to achieve reliable performance with fewer parameters while preserving the interpretability that capsule networks provide. The concepts and mechanisms introduced here lay the groundwork for deeper integration with the U-Net framework, explored later in this work through a three dimensional U-Net applied to rectum segmentation on low field MRI—a critical step toward improving image registration in prostate biopsy workflows.

4.1 Introduction

A Capsule Network, often referred to as CapsNet, is an advanced type of neural network that employs neuron clusters known as “capsules”. Unlike traditional Convolutional Neural Networks (CNNs), which output scalar values, these capsules produce vector outputs. These vectors represent not only the probability of a feature’s existence but also its instantiation parameters, such as pose (position, size, orientation), deformation, texture, and so on. This richer representation allows the network to capture and maintain spatial relationships between features more effectively than traditional methods. CapsNet introduces a unique mechanism, known as “routing-by-agreement”, which replaces the pooling layers found in conventional CNNs. This routing process enables capsules at one level to send their outputs to higher-level capsules only if there is a strong agreement (i.e., high probability) that the higher-level capsule’s entity is present in the input. This agreement is determined according to the dot product between the output of a lower-level capsule and the predicted output of a higher-level capsule, iteratively refined through a routing process. This architecture ensures that, during forward propagation, information flows through the network in a way that preserves spatial relationships, making it inherently more capable of handling variations in viewpoint, scale, and rotation without the need for extensive data augmentation [41].

Capsule networks aim to address some fundamental limitations of CNNs, especially in terms of preserving spatial hierarchies between features within an image. As CNNs rely on the scalar output of neurons within layers for feature detection and representation,

they sometimes fail to recognize objects captured from different viewpoints if they are not covered in the training data [42, 43]. CapsNets, in contrast, can better preserve the pose information (position and orientation) of features, thus making them more robust to variations in the input data [41, 44, 45].

There are several main distinctions between CapsNets and CNNs. First, CapsNets use a vector or group of neurons as a basic unit, whereas CNNs use a single neuron. These vectors, called capsules, can potentially represent different parts of an object. Second, while CNNs capture hierarchical features through the depth of the network, with each layer learning different levels of abstract and complex features, CapsNets model hierarchical relationships between parts and whole objects, providing more interpretable representations of learned features. Third, unlike CNNs—which lack a dedicated mechanism for routing information between layers—CapsNets use data-dependent routing to determine the flow of information between capsules, allowing for better modeling of part-whole relationships.

In classic routing procedures, a CapsNet begins with a set of primary capsules that represent low-level features extracted from the input data. Each primary capsule makes predictions (or votes) by computing an affine transformation of its output and sending their votes to the capsules of the next layer. These capsules at the higher level compute a weighted sum of the predictions received from capsules at the lower level. Routing weights are normalized through layer normalization [46] or a squashing function [41]. An iterative routing process is used to determine an agreement on how capsules at one level should connect to capsules in the next level by updating the weights [41, 44]. In contrast, a non-iterative routing procedure computes the routing

weights and information of capsules only once [45, 47]. By simplifying the process into a single forward pass, non-iterative routing methods alleviate the computational load associated with iteration.

Unlike conventional routing methods [41, 44], the cluster routing paradigm involves capsules generating vote clusters (instead of individual votes) for the subsequent layer’s capsules [47]. Each vote cluster consists of multiple votes, with each vote potentially originating from a distinct capsule in the previous layer. The proximity of votes within a cluster signifies the information extracted from the same part of an object from capsules in previous layers. Consequently, the variance within a vote cluster can serve as an indicator of the confidence in the encoded information that the vote clusters represent. This suggests that vote clusters with lower variance are more reliable in encoding information related to a specific part of an object. Thus, greater weights are assigned to centroids originating from clusters with lower variance.

We offer a comprehensive explanation of the definitions and terminologies presented in the existing literature [47]. Our objective is to minimize confusion and prevent misinterpretations surrounding the cluster routing technique, thereby enhancing its utilization across different scenarios.

Moreover, we carry out extensive experiments to investigate the impact of agreement indicators and hyperparameters. Our goal is to determine whether variance can serve as the sole dependable agreement indicator, or if increasing the number of votes per cluster invariably results in better performance.

Expanding upon the non-iterative cluster routing concept [47], we propose two variations of cluster routing: one features a streamlined routing process with the

potential to reduce the number of parameters while maintaining performance levels comparable to those of existing methods, and the other is designed to effectively handle an increase in the number of votes per cluster, thus improving the overall efficiency of cluster routing.

Empirical evaluations across several data sets show that our proposed methods utilizing cluster routing consistently yield higher accuracy, when compared to existing capsule networks. Moreover, our capsule network maintains the beneficial attributes of previous CapsNet versions, including strong generalization to images taken from new viewpoints.

The contributions of our work are outlined in the following:

- The definitions and examining the implementations within the cluster routing literature are clarified to support the effective deployment of cluster routing techniques.
- Comprehensive experiments are performed to investigate the impacts of agreement indicators.
- Two novel variations of cluster routing are proposed [47] based on the routing principle of achieving consensus among votes of clusters.

4.2 Related Work

4.2.1 Agreement Routing in Capsule Network

CapsNets play a crucial role in feature encoding. They transform the extracted features into capsules, which are vectors capable of characterizing parts of an object, unlike the singular units used in conventional CNNs. These capsules possess the ability to represent more intricate aspects of an object, such as its pose, orientation, and texture style. The capsule system proposed in the dynamic routing paper titled “Dynamic Routing Between Capsules” [41] employs a routing-by-agreement mechanism. This mechanism iteratively determines its predictions based on the agreement between lower- and higher-level capsules, which is achieved through adjusting the weights connecting them. Inspired by the ideas presented in this paper, numerous subsequent research papers have explored similar concepts with the aim of enhancing the capsule network while preserving the critical notion of the part–whole relationship. These endeavors have led to diverse approaches and innovations in the field. The matrix capsules with EM routing approach [44] uses high-dimensional coincidence filtering and a fast iterative process to determine the routing weights, resulting in better performance and higher robustness to adversarial attacks than baseline CNNs. RS-CapsNet [48] integrates *Res2Net* and *Squeeze-and-Excitation* blocks to extract multi-scale features, emphasizes useful information, and employs linear combinations between capsules in order to enhance object representation while reducing the capsule count. Cluster routing [47] uses variance as the fundamental indicator of agreement and confidence in the information encoded within the vote cluster. Group normalization [49] also

utilizes the mean and standard deviation of a group. It splits the output channels of a convolutional layer into several groups and normalizes the features within each group according to the mean and standard deviation. However, as it does not use routing weights based on the agreement between capsules (data-dependent routing) to produce the input for the next capsule layer, as in cluster routing, it is not classified as a routing algorithm.

CapsNets can outperform conventional CNNs in various applications, utilizing fewer parameters [50, 51, 52]. This advantage is particularly notable in medical image processing, where the main obstacles include detecting small lesions and overcoming class imbalances. Unlike CNNs, which often require substantial amounts of labeled data (that may be scarce in medical contexts), CapsNets can achieve similar levels of performance with a smaller data set [53, 54]. CapsNets employ capsules that encapsulate richer feature vectors, unlike CNNs that use scalar neurons, making them more effective in addressing these challenges [53]. Furthermore, CapsNets are superior in maintaining part-whole relationships and geometric details, significantly improving their performance in medical segmentation tasks [55, 51, 56, 52].

AI-generated deepfakes, including face-swapping videos and images, have been proliferating across the internet, driven by significant advancements in graphics processing units and AI algorithms. These technologies enable individuals to effortlessly create manipulated and unethical media. In areas such as deepfake detection, where novel and unforeseeable attacks are frequent, the strong generalization capability of capsules becomes vital. The nature of deepfake technology allows attackers to continuously develop new methods to bypass detection systems, making it imperative

for defense mechanisms to possess the ability to generalize from known attacks to novel ones effectively. CapsNets are particularly suited to this task, due to their ability to understand the underlying structure of the data in a way that mirrors human visual perception. This understanding includes recognizing when an image or video deviates from the norm in a manner that suggests manipulation, even if the specific technique used for manipulation has not been encountered by the system before. A notable application of CapsNets in this context is Capsule-Forensic, which has been shown to be effective in identifying altered or synthetically produced images and videos [57, 58]. The efficacy of CapsNets in this application stems from their unique ability to encode hierarchical relationships between objects and their components, including detailed pose information. This makes it an invaluable tool in the fight against the unethical use of AI for media manipulation [59].

4.2.2 Attention Routing

The ideas behind CapsNets share similarities with those of attention mechanisms, initially introduced in transformers [60]. Attention between capsules [61] replaces dynamic routing with a convolutional transform and attention routing, thus utilizing fewer parameters. The inclusion of a dual attention mechanism after the convolution layer and primary capsules in [62] enhanced the performance of the CapsNets. The use of a self-attention mechanism in [63] allowed for alternative non-iterative routing, efficiently reducing the number of parameters while maintaining effectiveness.

4.3 Methods

4.3.1 Notation and Terminology

Several terms and definitions are used in the relevant literature, defined as follows:

- $\mathbf{v}$, *Vote*: A vote is the prediction of a capsule, resulting from the multiplication of a weight matrix with a capsule.
- $\mathbf{D}$, *Depth or Length of a Capsule*: The number of convolutional units in a capsule.
- $\mathbf{u}$, *Capsule*: A capsule with length or depth of $\mathbf{D}$.
- $\mathbf{K}$, *Number of Votes per Cluster*: Each cluster has $\mathbf{K}$ votes.
- $\mathbf{N}$, *Number of Clusters*: The number of vote clusters in each channel of a capsule layer.
- $\mathbf{W}$, *Weight Matrix*: A weight matrix is multiplied with a capsule to create a vote $\mathbf{v}$. There are $\mathbf{K}$ matrices to produce $\mathbf{K}$ votes in a cluster.
- $\mathbf{C}$, *Number of Channels*: Each capsule layer has $\mathbf{C}$ channels.
- $\mathbf{G}$, *Number of Groups*: Each group has a number of clusters.

In cluster routing, a capsule layer has $\mathbf{C}$ channels, each channel has $\mathbf{N}$ clusters, and each cluster has $\mathbf{K}$ votes.

4.3.2 Original Non-Iterative Cluster Routing

Cluster routing [47] generates clusters of votes through a non-iterative approach, where the standard deviation is employed as the fundamental indicator of agreement and confidence for the information encoded within the vote cluster. Clusters with higher variance indicate lower confidence in their encoded information or decision; consequently, they are weighted less and passed to the next layer. The agreement vector is computed based on the standard deviation of the cluster's individual votes. The centroids or means of the vote clusters are weighted with their corresponding agreement vectors. The weighted sum of these centroids integrates and aggregates the information from all vote clusters across the capsules in the current layer, which is then normalized and transmitted to the next layer. By leveraging agreement and weighted summation, cluster routing can efficiently and dynamically route information flow between each layer of capsules in a single pass. A vote cluster $\hat{\mathbf{u}}_i^{\{1, \dots, K\}}$ can be generated by multiplying a capsule $\mathbf{u}_i$ with each weight matrix $\mathbf{W}$ of a weight cluster $\mathbf{W}_i^{\{1, \dots, K\}}$,

$$\hat{\mathbf{u}}_i^k = \mathbf{W}_i^k \mathbf{u}_i. \quad (4.1)$$

To enhance the receptive fields of each vote, the neighborhood of $\mathbf{u}_i$ in Equation (4.1) is expanded to 3×3 in practice. $\mathbf{u}_i$ becomes the concatenation $N(\mathbf{u}_i) \in \mathbb{R}^{9D}$ of its neighborhood. Then, the k th vote in the cluster is produced as follows,

$$\hat{\mathbf{u}}_i^k = \mathbf{W}_i^k N(\mathbf{u}_i). \quad (4.2)$$

A vote cluster $\hat{\mathbf{u}}_i^{\{1, \dots, K\}}$ will send its centroid or mean $\mathbf{m}_i = \frac{1}{K} \sum_{k=1}^K \hat{\mathbf{u}}_i^k$ to the next layer, with weighting determined by the agreement vector $\mathbf{a}_i$. The computation of $\mathbf{a}_i$

is outlined as follows,

$$\mathbf{a}_i = -\log(\sigma_i), \quad (4.3)$$

where

$$\sigma_i = \sqrt{\frac{1}{K} \sum_{k=1}^K (\hat{\mathbf{u}}_i^k - \mathbf{m}_i^k) \times (\hat{\mathbf{u}}_i^k - \mathbf{m}_i^k)} \quad (4.4)$$

The centroids (or means) of these vote clusters are weighted summed as follows,

$$\mathbf{s} = \sum_{i=1}^C \mathbf{c}_i \times \mathbf{m}_i, \quad (4.5)$$

where

$$\mathbf{c}_i = \frac{\exp(\mathbf{a}_i)}{\sum_{j=1}^C \exp(\mathbf{a}_j)} \quad (4.6)$$

Each of the $\mathbf{C}$ capsule channels contributes a vote cluster to the capsules in the next layer, making the number of vote clusters ($\mathbf{N}$) equivalent to the number of channels ($\mathbf{C}$) in the multiple-channel version. Layer normalization [46] is applied to $\mathbf{s}$ to produce the output of the capsule layer. The computational task in Equation (4.2) can be efficiently executed through utilization of PyTorch’s pre-existing optimized convolutional operations [64], similar to the technique employed in [61].

Notably, this mechanism utilizes significantly fewer parameters than both attention routing [61] and dynamic routing [41]. The absence of iterations in the process enhances its computational efficiency and reduces training resource requirements.

4.3.3 Cluster Routing Implementation

Weighting Computation

Feature maps were extracted from the original image through a convolutional layer to form primary capsules. This layer is a convolutional capsule layer with $\mathbf{C}$ channels of

convolutional capsules. Each primary capsule comprises $\mathbf{D}$ convolutional units with a 3×3 kernel and a stride of 1. To create a vote, cluster routing utilizes a weight matrix for element-wise multiplication with a capsule. Capsules within the same channel can share weights to produce a vote, where each capsule has dimensions of $1 \times 1 \times \mathbf{D}$. However, in cluster routing, the surrounding neighborhood is included to increase the receptive field for producing a vote, resulting in the weight dimension of a capsule becoming $3 \times 3 \times \mathbf{D}$.

To handle the weight multiplication process, we leverage an optimized function in PyTorch named `conv2d`. There are two approaches to produce votes from $\mathbf{C}$ separate channels in the primary capsule layer.

First, a single `conv2d` layer can be used to generate votes from all $\mathbf{C}$ channels, employing group convolution through the `conv2d` function in PyTorch. The `groups` parameter in `conv2d` controls the connections between inputs and outputs; for example, if `groups = 1`, all inputs are convolved to all outputs. In cluster routing with $C = 3$, we can use `groups = 3` to consider three convolutional layers side by side, creating three capsule channels. Each layer sees one-third of the input tensor and produces one-third of the output, which are subsequently concatenated.

Second, the feature maps can be segmented into individual channels, with each channel employing its own `conv2d` layer to generate votes. It is important to note that each channel utilizes a distinct `conv2d` layer for the generation of votes, which is currently implemented in the code.

In iterative routing [41, 44], the routing weights are computed through an iterative process. The weight update frequency can have significant effects on the performance

[41]. However, in our non-iterative approach, the weight only needs to be computed once per epoch based on the agreement vectors. Hence, the weight update frequency does not impact the performance. In fact, without iteration, our method can reduce the training time to approximately 15%.

Agreement Vector

In cluster routing, the weighting coefficients c_i in Equation (4.6) can be computed efficiently using the `softmax` function in Pytorch. However, when we substitute Equation (4.3) into Equation (4.6), the c_i can be simplified into the following form,

$$\mathbf{c}_i = \frac{\exp(\mathbf{a}_i)}{\sum_{j=1}^C \exp(\mathbf{a}_j)} = \frac{\frac{1}{\sigma_i}}{\sum_{j=1}^C (\frac{1}{\sigma_j})} \quad (4.7)$$

where σ_i is the standard deviation of the vote cluster.

The agreement vector in Equation (4.6) now can be redefined as $\mathbf{a}_i = \frac{1}{\sigma_i}$. This can be explained by stating that the agreement is inversely proportional to the standard deviation of the vote cluster: with a higher standard deviation, less agreement exists between the votes, resulting in less weight being assigned to the centroid of the cluster.

Cluster Routing with Single Channel

In cluster routing, optimal performance is typically achieved through utilization of the single-channel version, as illustrated in Table 4.1. Technically, this approach can be summarized as creating multiple versions of the original capsule channel, increasing its dimension by one. Next, the `.view()` function is utilized to divide the additional dimension into two dimensions, $\mathbf{K}$ and $\mathbf{N}$, as depicted in Figure 4.1. Subsequently, the mean and standard deviation are computed along the $\mathbf{K}$ dimension

to determine the centroid and agreement vector, reducing the dimension by one. Following this, the mean tensor is multiplied with the weight and then summed to derive the output capsule channel, further reducing the dimension by one. This process involves expanding the dimension and subsequently reducing it to preserve the original dimensionality of the input capsule channel. Further details and the code implementation of this version can be found in our GitHub repository.

The averaging and summing components of this technique can be conceptualized as a type of vector dimension reduction, in which the dimensionality is reduced while preserving essential information. However, additional experiments (which are not addressed in this study) are required to validate this hypothesis.

Fig. 4.1: Comparative analysis of test error rates in the capsule networks literature [47].

Method	smallNORB		Fashion-MNIST		SVHN		CIFAR-10	
	Error (%)	Param	Error (%)	Param	Error (%)	Param	Error (%)	Param
Baseline CNN	3.76	3.30M	5.21	3.38M	3.30	3.38M	7.90	3.38M
Dynamic [41]	2.7	8.2M	-	-	4.3	$\simeq 1.8$ M	10.6	8.2M
EM-Routing [44]	1.8	310K	-	-	-	-	11.9	$\simeq 460$ K
RS-CapsNet [48]	-	-	6.49	5.00M	3.5	5.01M	10.19	5.01M
Attention Routing [61]	-	-	-	-	-	-	11.39	9.6M
DA-CapsNet [62]	1.74	-	6.02	-	5.18	-	14.53	-
Efficient-CapsNet [63]	2.54	151K	-	-	-	-	-	-
RVC [65]	-	-	5.87	-	2.97	-	8.3	-
Inverted dot-product	-	-	-	-	-	-	14.83	560K
attention routing [66]	-	-	-	-	-	-	-	-
VB-Routing [67]	1.6	169K	5.2	172K	3.9	323K	11.2	$\simeq 323$ k
MoCapsNet [68]	-	-	-	-	7.00	-	27.83	-
SRM-CapsNet [69]	8.54	-	8.35	-	6.90	-	28.26	-
ER-CapsNet [70]	-	-	-	-	4.33	-	11.46	9.4M
Metric Learning [71]	-	-	9.59	-	7.24	-	26.76	9.4M

Fig. 4.1: *Cont.*

Method	smallNORB		Fashion-MNIST		SVHN		CIFAR-10	
	Error (%)	Param	Error (%)	Param	Error (%)	Param	Error (%)	Param
MASK DR [72]	–	–	6.32	1.4M	–	–	9.99	1.4M
ML-CapsNet [73]	–	–	7.35	–	–	–	14.28	–
PT-CapsNet [74]	–	–	7.77	–	–	–	8.79	–
NASCaps [75]	–	–	6.13	–	3.41	–	23.54	–
DenseCaps [76]	–	–	–	–	4.01	–	10.59	16M
<i>Cluster Routing</i> [47]								
CM^1_{456} (C4K5D6)	2.98±0.24	150K	5.17±0.07	146K	3.94±0.07	154K	12.16±0.30	154K
CM_{458} (C4K5D8)	3.09±0.19	246K	5.02±0.04	240K	3.63±0.11	252K	11.11±0.09	252K
CM_{4816} (C4K8D16)	1.92±0.12	1.32M	4.84±0.07	1.30M	3.56±0.07	1.34M	8.55±0.12	1.34M
CM_{4824} (C4K8D24)	1.95±0.12	2.87M	4.64 ±0.03	2.84M	3.48±0.14	2.89M	7.89±0.11	2.89M
CS^2 (N8K8D32)	1.57 ±0.13	2.53M	4.68±0.01	2.51M	3.37 ±0.03	2.55M	7.37 ±0.06	2.55M
<i>Ours</i>								
$V1M^3_{456}$ (K20D6G1)	3.32±0.46	150K	5.36±0.07	146K	3.74±0.03	154K	10.78±0.34	154K
$V1M_{458}$ (K20D8G1)	3.11±0.41	246K	5.29±0.08	240K	3.71±0.06	252K	10.21±0.24	252K
$V1M_{4816}$ (K32D16G4)	2.67±0.30	1.32M	5.06±0.04	1.30M	3.62±0.14	1.34M	8.22±0.30	1.34M
$V1M_{4324}$ (K12D24G8)	1.95 ±0.43	1.11M	4.95±0.03	1.09M	3.48 ±0.08	1.11M	7.83 ±0.12	1.11M
$V1M_{4824}$ (K32D24G8)	2.10±0.22	2.87M	4.86±0.02	2.84M	3.73±0.14	2.89M	7.53±0.11	2.89M
$V2S^4$ (N4K2N4K2D32)	1.46 ±0.11	2.53M	4.63 ±0.07	2.51M	3.21 ±0.08	2.55M	7.33 ±0.03	2.55M

¹ Cluster routing with multiple channels. ² Single channel. ³ First variation

(multi-channel). ⁴ Second variation (single channel).

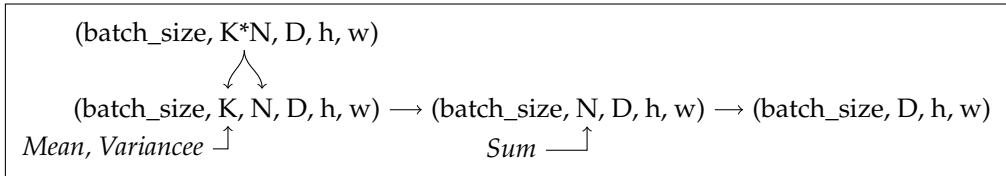


Fig. 4.1: Single-channel quick implementation of cluster routing.

Cluster Routing with Multiple Channels

Cluster routing can also be implemented with multiple channels. Each capsule layer consists of $\mathbf{C}$ channels and, within each channel, there are $\mathbf{N}$ clusters. As each channel contributes a vote cluster, there are $\mathbf{C}$ vote clusters per channel, making $\mathbf{N}$ equal to $\mathbf{C}$ in this case. For simplicity, this variation involves multiple channels, with each channel representing the single-channel version of cluster routing, as previously presented.

4.3.4 Proposed Routing Variations

Our First Routing Variation

This deviated version originates from the multiple-channel version of cluster routing. Nevertheless, it remains compatible with the single-channel version, and we adhered to the agreement based on standard deviation. However, we omit the step of summation over the weighted centroids, thereby eliminating the need for the hyperparameter $\mathbf{N}$, representing the number of clusters per channel. The process is illustrated in Figure 4.2.

The agreement vector $\mathbf{a}_i$ in Equation (4.3) is simplified as the reciprocal of its standard deviation, as follows:

$$\mathbf{a}_i = \frac{1}{\sqrt{\frac{1}{K} \sum_{k=1}^K (\hat{\mathbf{u}}_i^k - \mathbf{m}_i^k)^2}} \quad (4.8)$$

Apart from the weighted sum in Equation (4.5), the centroids of these vote groups are only weighted and fast-forwarded to the next layers, as follows,

$$\mathbf{s} = \mathbf{c}_i \times \mathbf{m}_i, \quad (4.9)$$

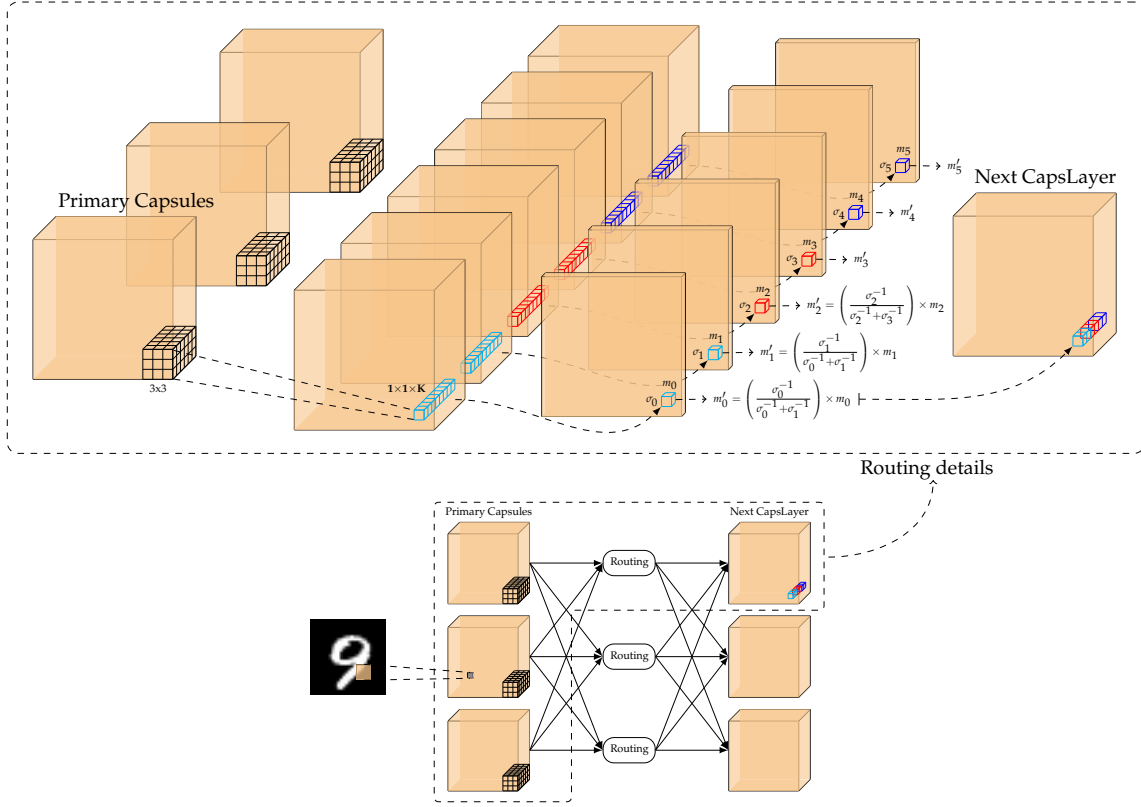


Fig. 4.2: Overview of the proposed multi-channel routing variation.

where,

$$\mathbf{c}_i = \frac{\mathbf{a}_i}{\sum_{j=1}^C \mathbf{a}_j} \quad (4.10)$$

Then, layer normalization [46] is applied to the weighted centroids $\mathbf{s}$. The fast-forward mechanism, when experimentally applied, exhibited comparable performance with fewer parameters than the weighted sum approach used in cluster routing for the *SVHN* and *CIFAR-10* data sets, as illustrated in Table 4.1. This method has the potential to reduce the number of parameters while preserving network performance.

To demonstrate the process, the bottom of Figure 4.2 illustrates the connections between the capsule layers. At the top of the figure, 3×3 primary capsules are multiplied with each weight matrix of a weight cluster comprising $\mathbf{K}$ matrices $\mathbf{W}_i^{\{1, \dots, K\}}$ to generate a vote cluster $\mathbf{u}_i^{\{1, \dots, K\}}$. This vote cluster consists of $\mathbf{K}$ votes, depicted as a $1 \times 1 \times \mathbf{K}$ block (colored cyan) in the right corner of the front middle block. Each cluster sends its weighted centroid $\mathbf{m}_i$ based on the agreement vector $\mathbf{a}_i$ to the next capsule layer. The agreement vector is the result of comparing the standard deviation of each vote cluster with other vote clusters within their smaller respective group, distinguished using different colors, where the number of groups is denoted as $\mathbf{G}$. In the current scenario, there are three groups ($G = 3$), each with two centroids (denoted as $\mathbf{m}_0$ and $\mathbf{m}_1$). These centroids are weighted using their corresponding standard deviations σ_0 and σ_1 , resulting in transformed values (labeled as $\mathbf{m}'_0$ and $\mathbf{m}'_1$ for the cyan-colored group). This procedure is replicated for the other (red and blue) groups. The resultant weighted centroids are concatenated together to form the next capsule channel layer.

Our Second Routing Variation

While conducting extensive experiments with various hyperparameters in cluster routing, we observed that increasing the number of votes per cluster (denoted as $\mathbf{K}$) in both single- and multiple-channel versions can lead to a decrease in performance, as demonstrated in Figures 4.3 and 4.4, respectively. Therefore, we introduce a second variation of the routing mechanism to mitigate the impact of increasing $\mathbf{K}$ while enhancing its performance.

In Figure 4.5, the $\mathbf{K}$ and $\mathbf{N}$ dimensions are split only once in the original cluster routing, using the `.view()` method in PyTorch. This second variation suggests that the breakdown is not limited to only two times (as depicted in Figure 4.5) but, rather, depends on the dimension of the tensor, which must satisfy the condition $K1 \times N1 \times K2 \times N2 = K \times N$.

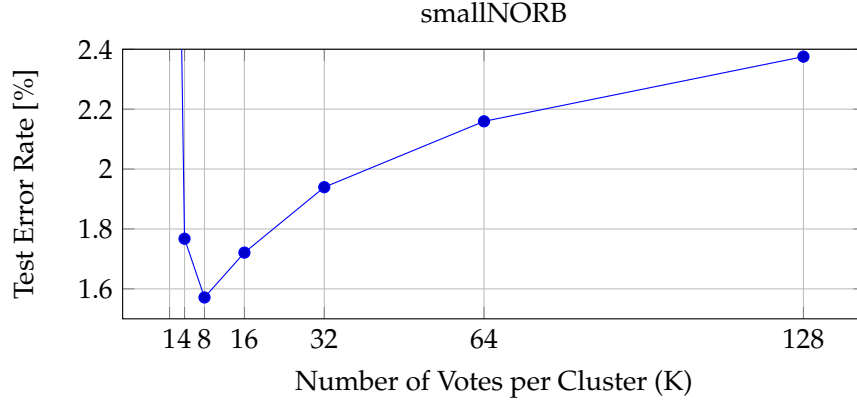


Fig. 4.3: Performance of the single-channel version of cluster routing with $\mathbf{N} = 8$ and $\mathbf{D} = 32$.

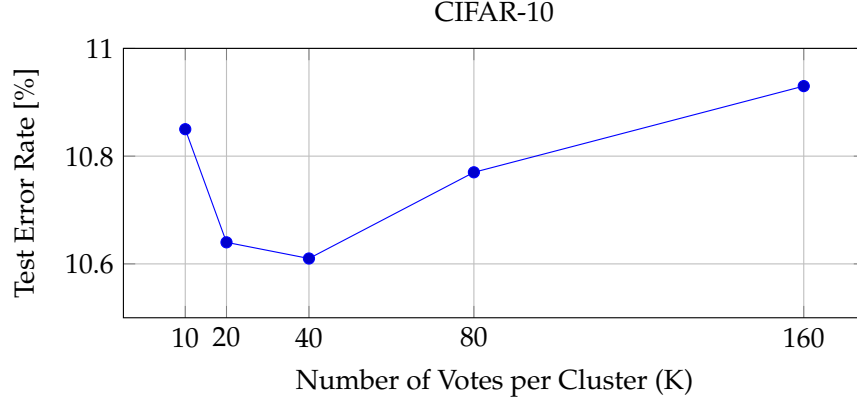


Fig. 4.4: Performance of the multiple-channel version of cluster routing with $\mathbf{C} = 4$ and $\mathbf{D} = 6$.

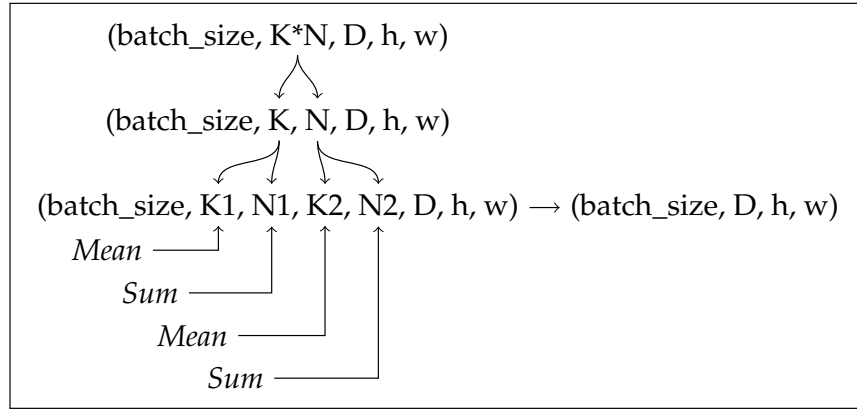


Fig. 4.5: Second variation scheme.

This approach can be utilized to effectively manage capsule layers with large dimensions, resulting in improved performance and faster convergence, as illustrated in Figures 4.6 and 4.7.

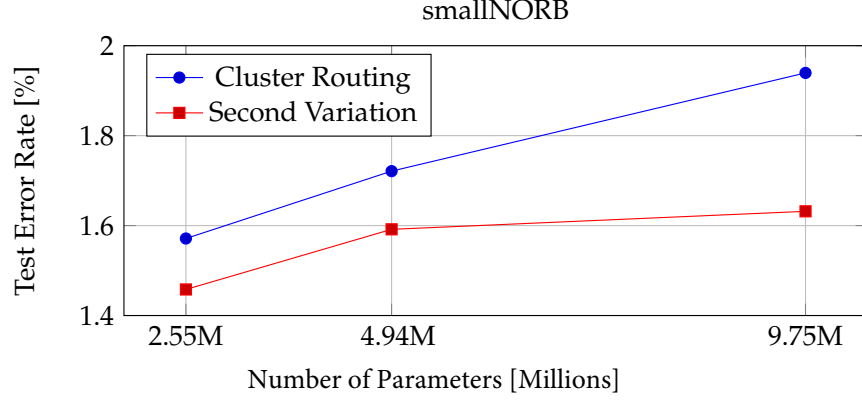


Fig. 4.6: Performance of cluster routing and the Second Variation.

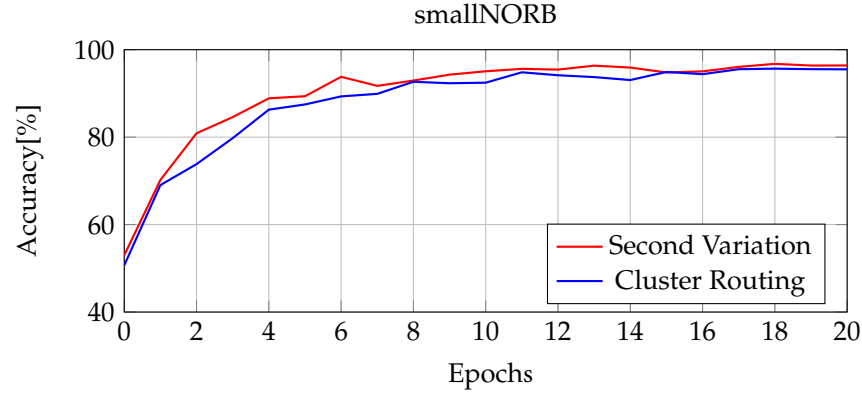


Fig. 4.7: Training curves for cluster routing and the second routing variation.

4.4 Experimental Setup

In all experiments, we set the number of capsule layers to 5 and used a stride of 2 in the second and fourth layers. The models were all trained with the Stochastic Gradient Descent optimizer and cross-entropy loss. Training was performed with a batch size of 64 over 300 epochs, with an initial learning rate of 0.1 and a step decay of 0.1 at every 100 epochs. The hyperparameters for augmentation were tuned using a validation set comprising one-fifth of the training images.

The reference CNN employed for comparison was composed of five ReLU convolutional layers. Each layer incorporates ReLU activation, 256 filters, and layer normalization, resulting in a total of 3.38 million parameters.

Data Pre-Processing

For each data set, we employed brightness and contrast jitter to augment the training data. We used α and β to control the contrast and brightness $f(x)$ of a pixel at position x by $g(x) = \alpha f(x) + \beta$. Specifically, random brightness and contrast were applied with a factor of 0.2. The value of α was randomly chosen from a uniform distribution in $[0.8, 1.2]$, and the value of β from $[-0.2 \frac{1}{N_x} \sum_x f(x), 0.2 \frac{1}{N_x} \sum_x f(x)]$, with N_x denoting the total number of pixels and $\frac{1}{N_x} \sum_x f(x)$ denoting the mean pixel value.

The *smallNORB* [77] data set contains 24,300 images at 96×96 resolution, classified into 5 classes. Pre-processing, as per [44], involved down-sampling to 48×48 , normalization and, for training, a contrast factor of 0.2, random shifts of 0.2, padding to 56×56 , and random cropping to 32×32 . For testing, images were centrally cropped to 32×32 .

The *Fashion-MNIST* [78] data set contains 70,000 images with a resolution of 28×28 pixels, categorized into 10 classes. The data set was split into training and testing sets with a ratio of 6:1. During training, various data augmentation techniques were applied to the images, including random brightness adjustments, random flips with a 0.5 probability, and contrast adjustments with a factor of 0.2. Additionally, the images were padded to a size of 36×36 pixels, then randomly cropped to a final

size of 32×32 pixels.

The *CIFAR-10* [79] data set consists of a collection of 60,000 color images, each measuring 32×32 pixels, and is divided into 10 different classes, with each class representing a distinct object or category. Each class contains 6000 images. The training and testing sets were divided with a 5:1 ratio. In the training phase, random adjustments to brightness and contrast were implemented with a factor of 0.2. Padding to achieve a size of 40×40 was conducted, then random cropping was applied to obtain a 32×32 portion. Additionally, random horizontal flips were introduced with a 0.5 probability.

The *SVHN* [80] data set comprises 10 digit classes represented as 32×32 real-world house numbers. The core training set, consisting of 73,257 images, was used for training, while evaluation was performed on the 26,032 test set images. Throughout training, random brightness and contrast adjustments with a factor of 0.2 were applied. Images were padded to 40×40 , and random 32×32 crops were extracted for further processing.

4.5 Results

In Table 4.1, we evaluate and compare the performance of our proposed capsule network in classification tasks (although it is not exclusively confined to such tasks). We investigated its generalization to novel viewpoints and experiments through the various indicators of agreement between cluster votes. For each data set, we conducted a minimum of 8 experiments using 8 random seeds under identical conditions, and

the average test error rate, standard deviation, and the number of parameters are presented in Table 4.1.

4.5.1 Comparison with the Baseline CNN

In our study, our proposed cluster routing variations demonstrated superior performance when compared to the baseline CNN in two key aspects: the number of parameters and test error rates. Notably, when tested on the smallNORB data set, which is known for its variations in viewpoint, the proposed variations achieved higher accuracy while employing fewer parameters, as illustrated in Table 4.1. The CNN baseline, with 3.3 million parameters, achieved a test error rate of 3.76% on the smallNORB data set, while our strategy halved the test error rate to 1.95% while using approximately one-third of the parameters (1.11 M). The proposed variation achieved the best performance on this data set, with a test error rate of 1.46%. This proves the robustness of CapsNets in handling viewpoint variations in object recognition and pose estimation tasks. This accomplishment can be attributed to the capsule network’s inherent capability to retain and utilize pose information within the features, enabling robust generalization even when confronted with images captured from viewpoints that have not been previously encountered [65]. While demonstrating superior performance and robustness on the smallNORB data set, our proposed variations also exhibited improved performance with a reduction in the number of parameters across image processing tasks, including the real-world digit recognition task in the SVHN data set, as well as image classification in the Fashion-MNIST and CIFAR-10 data sets, as illustrated in Table 4.1. This can be useful in real-world scenarios, where objects may

be partially occluded or where the viewpoint of the object may vary.

CNNs are adept at a wide array of tasks, but encounter difficulties in maintaining the spatial relationships between objects due to the incorporation of max-pooling layers [53, 55, 81]. These layers, which are designed for feature extraction and dimensionality reduction, inadvertently result in the loss of spatial information among objects. Consequently, CNNs may demonstrate decreased robustness in scenarios involving changes in viewpoint, scale, and rotation [54]. Capsule networks offer a solution to this issue by eliminating the need for pooling layers. Instead of relying on scalar values for feature representation—as is common in CNNs—CapsNets employ vectors or “capsules”. Furthermore, they utilize a “routing-by-agreement” mechanism instead of pooling, which allows capsules to collaboratively decide on the existence of higher-level features [50]. This capability to preserve spatial relationships renders CapsNets particularly beneficial in the field of medical image analysis for tasks such as tumor detection, semantic segmentation, and disease classification, especially considering the variability in the body shapes and sizes of patients, as well as tumor locations.

Despite their potential, CapsNets face inherent challenges; notably, they depend on a sophisticated routing algorithm that requires extensive agreement computations among capsules. In seminal CapsNet studies, such as those cited by [41, 44], routing iterations have been shown to be crucial for updating capsule agreements, considerably lengthening training and inference times [50]. This issue could limit the ability of CapsNets to match the depth and scalability of CNNs [52]. Nevertheless, our novel approaches leverage a non-iterative strategy to calculate capsule agreement, effectively

reducing the training duration by an average of 15%, when compared to conventional iterative methods [41, 44].

4.5.2 Comparisons with the State-of-the-Art

For a fair comparison in Table 4.1, we employed identical data processing and training settings, ensuring the selection of configurations with a similar number of parameters.

Our first variation, characterized by a lower number of parameters, achieved performance comparable to that of the original cluster routing approach. Specifically, the *V1M4324* variation with fewer parameters (1.11 million in total) demonstrated a test error rate of 3.48% on the *SVHN* data set. This is comparable to that of the original cluster routing approach, which has 2.89 million parameters and a similar error rate. Similarly, in the context of the *CIFAR-10* data set, the outcomes of the *V1M4324* variation closely matched those of the original cluster routing, despite employing fewer parameters, with test error rates of 7.83% and 7.89%, respectively. However, it slightly fell short in performance on the *Fashion-MNIST* data set, exhibiting a test error rate of 4.95% with 1.1M parameters, in contrast to the original cluster routing, which achieved 4.84% with 1.3M parameters. Despite this result, this variation demonstrated the potential to achieve comparable performance with fewer parameters. Furthermore, our variation employed a simpler agreement calculation, leading to a modest reduction in training time (averaging 5%), when compared to the original cluster routing approach.

Our second variation model, *V2S*, demonstrated significant performance enhancements compared to existing capsule networks with an equivalent number of parameters to the original cluster routing, as detailed in Table 4.1. Our analysis revealed that

an increase in the number of votes per cluster, denoted as $\mathbf{K}$, led to a degradation in the performance of the original cluster routing in both single- and multi-channel configurations, as depicted in Figures 4.3 and 4.4, respectively. The *V2S* model addresses this issue effectively, offering a strategic approach to manage capsule layers with significant dimensions and alleviate the negative impact of increasing $\mathbf{K}$. In the context of the *smallNORB* data set, *V2S* not only achieved superior performance, as demonstrated in Figure 4.6, but also exhibited faster convergence during network training, as illustrated in Figure 4.7. These characteristics significantly enhance the cluster routing efficacy, underscoring the potential of the *V2S* model in improving cluster routing capabilities.

Despite nuanced variations, the two proposed routing alternatives emerge as viable options, showcasing notable strengths in differentiating data sets with diverse characteristics.

4.5.3 Classification across Various Viewpoints

In this section, we evaluate the generalization capabilities of our proposed methods to novel viewpoints using the *smallNORB* data set. To guarantee a fair comparison, we also retrained the cluster routing algorithm, applying the same data augmentation techniques as those used in our methods.

The network was trained on the training set with varying capture angles or azimuths (300, 320, 340, 0, 20, 40). The test set comprised different instances in the same categories, with unseen angles ranging from 60 to 280. To evaluate the network’s performance under varying elevation viewpoints, we selected three smaller elevations

from the training set and tested the model on six larger elevations from the test set.

From the novel azimuth viewpoint, our first variation achieved the highest accuracy of 85.27%. The multiple-channel version of the original cluster routing and our first variation demonstrated an advantage over the single-channel version. Regarding the novel elevation viewpoint, the single version of our second variation attained the highest accuracy of 84.44%, as depicted in Table 4.2. The performance of both variations was on par with the original cluster routing, maintaining robust generalization to novel viewpoints. In certain instances, they even exhibited a slight advantage.

Fig. 4.2: Comparison of generalization capabilities on the *smallNORB* dataset across various viewpoints.

Model	Azimuth (%)				Elevation (%)			
	CS	CM	V2S	V1M	CS	CM	V2S	V1M
#Params	2.55M	2.89M	2.55M	2.89M	2.55M	2.89M	2.55M	2.89M
Novel View	83.31	84.89	83.84	85.27	83.79	84.25	84.44	83.71

4.5.4 Agreement Indicators

In cluster routing, the standard deviation of the votes within a cluster serves as a metric for assessing the consensus among votes surrounding the cluster centroids. Specifically, $\frac{1}{\sigma}$ is utilized to calculate the agreement weighting for the cluster centroids in cluster routing.

We performed experiments to investigate alternative agreement indicators, including $\frac{1}{\sqrt{\sigma}}$, $\frac{1}{\sigma^2}$, $\frac{1}{\sigma^3}$, and the mean of absolute difference. However, no significant

variations in network performance were observed. Additionally, we explored alternative indicators, such as the L1 norm (Manhattan norm) and the L2 norm (Euclidean norm), utilizing a moving 3D kernel for norm calculations. Despite these efforts, such alternatives led to a decrease in performance.

The summarized performance results for these investigations are provided in Table 4.3. Based on these findings, we concluded that $\frac{1}{\sigma}$ stands out as the optimal choice for the agreement indicator, due to its simplicity and efficiency.

Fig. 4.3: Performance of *second variation* on CIFAR-10 with diverse agreement indicators.

$\sqrt{\sigma}$	σ	σ^2	σ^3	MAD *	L1 Norm	L2 Norm
10.92 ± 0.33	10.78 ± 0.34	10.75 ± 0.37	11.05 ± 0.54	11.26 ± 0.39	34.41 ± 0.16	29.44 ± 0.25

* Mean of absolute difference.

4.6 Conclusions

This research aimed to clarify the key definitions and concepts integral to cluster routing, with an emphasis on demystifying the terminologies and principles underlying the original cluster routing algorithm. Additionally, a comprehensive guide was provided for the implementation of different versions of the original cluster routing framework, with the goal of streamlining its deployment and facilitating broader adoption across diverse applications.

Extensive experiments were conducted to assess the limitations associated with increasing the number of votes per cluster and to explore alternative agreement indicators for cluster routing. The findings indicated that the standard deviation of the votes within a cluster stands out as the optimal choice for the agreement indicator,

owing to its simplicity and efficiency. It was also observed that increasing the number of votes per cluster, denoted as $\mathbf{K}$, led to decreased performance in the original cluster routing for both single- and multiple-channel configurations.

Furthermore, two innovative variations of the cluster routing mechanism were proposed to enhance its capabilities. The first variation simplifies the original computation process by omitting the summation over weighted centroids, thereby reducing training time while maintaining comparable performance with fewer parameters. This simplification is advantageous, as computational complexity and memory requirements often constrain the design of deeper and larger capsule networks. The second variation addresses the impact of increasing the number of votes per cluster by enabling capsule layers to effectively handle higher-dimensional representations. This approach improves performance relative to the original cluster routing and offers faster convergence during network training.

Application to 3D Volumetric Segmentation

To further evaluate the practicality of the proposed non iterative cluster routing mechanism, it was integrated into a 3D segmentation framework based on the 3D-UCaps architecture [82]. In this experiment, the dynamic routing component of the bottleneck layer was replaced with the proposed cluster routing algorithm, while preserving the overall U-Net structure and the number of parameters. The modified architecture was trained on volumetric medical image data under the same experimental conditions as the original 3D-UCaps baseline.

Quantitative evaluations revealed that the cluster routing variant achieved seg-

mentation accuracy comparable to the dynamic routing baseline, with only marginal differences in Dice similarity coefficient (DSC) and average surface distance (ASD). While the non iterative routing reduced training time by eliminating iterative weight updates, it did not yield a substantial improvement in segmentation performance. This limited gain is attributed to the architectural design of 3D-UCaps, where the capsule routing mechanism is confined to a small bottleneck region, exerting limited influence on global feature propagation throughout the encoder–decoder hierarchy.

These results indicate that achieving noticeable improvement in segmentation tasks requires more than substituting the routing algorithm within an existing architecture. To fully realize the potential of capsule networks in segmentation, the structural design of the U-Net itself must be reimaged. A dramatic architectural shift—where capsule layers replace or augment multiple encoder and decoder blocks, and routing mechanisms operate across scales—is necessary to allow capsules to capture both local and global spatial dependencies. Such an approach would enable agreement-based routing to influence the entire feature flow rather than a restricted bottleneck, leading to a more coherent and context-aware segmentation process.

Future Directions

The proposed methods improved the effectiveness and interpretability of cluster routing, demonstrating potential for broader applications beyond classification tasks. Future work will focus on restructuring the U-Net architecture to integrate capsule routing more deeply across multiple layers and scales, enabling a unified capsule-based segmentation framework. In addition, these approaches will be explored in domains

such as medical image analysis, volumetric segmentation, and deepfake detection. The continued development of non iterative capsule routing mechanisms may contribute to more efficient and scalable deep learning models capable of preserving spatial hierarchies while reducing computational overhead.

Chapter 5

Enhancing Semantic Segmentation with Reinforced Active Learning

Deep learning based segmentation models such as U-Net have achieved remarkable accuracy across medical and natural imaging domains. Their encoder–decoder structures enable the extraction of rich hierarchical features that integrate both local and global context, advancing the state of semantic segmentation beyond handcrafted methods. However, these gains come with significant dependence on large, well-annotated datasets. In medical imaging, where expert annotation often requires trained radiologists or other specialists, the process is expensive, time-consuming, and prone to inconsistency. Questions of how much annotation is sufficient, which samples should be labeled first, and how annotation order influences model performance remain open and critical challenges. Even with powerful architectures, dataset imbalance and limited labeled data continue to hinder generalization, especially in clinical environments where data diversity is constrained by patient demographics and ethical considerations.

The motivation of this chapter stems from these challenges: how to train high-performing segmentation systems efficiently when annotation is costly and data are scarce. Instead of increasing dataset size indefinitely, a more intelligent strategy is to make the annotation process itself adaptive and data-driven. Reinforced active learning offers such an approach by allowing the model to learn what to annotate next, balancing exploration of new samples with exploitation of those most informative for improving performance. Through reinforcement learning principles, the selection

policy evolves based on prior annotation outcomes, thereby reducing labeling effort while accelerating convergence.

This chapter therefore explores reinforced active learning for semantic segmentation, focusing on how the integration of reinforcement learning strategies can optimize both annotation efficiency and model robustness. It aims to address fundamental questions about annotation sufficiency, selection order, and efficient training under limited supervision—paving the way for practical, scalable, and cost-effective segmentation frameworks suitable for real-world medical applications.

5.1 Introduction

Semantic segmentation assigns a class label to each pixel in an image, dividing it into meaningful regions that help interpret visual information at a detailed level. Unlike image classification, which gives a single label to an entire image, semantic segmentation provides fine pixel-level understanding that allows for more precise analysis.

Training large segmentation models, however, faces two main challenges: dataset imbalance and the heavy cost of annotation. In clinical datasets, differences related to age or gender, as well as uneven representation in natural images, can strongly influence model performance. In addition, producing pixel-level annotations is both time-consuming and labor-intensive. This study addresses these issues through reinforced active learning, which combines methods such as Dueling Deep Q-Networks, Prioritized Experience Replay, Noisy Networks, and Emphasizing Recent Experience to improve

both efficiency and accuracy. The approach aims to speed up annotation by selecting the most informative and representative samples while reducing dataset bias through careful learning strategies.

The goal of this research is to build a high-performing and reliable framework for accurate semantic segmentation, especially when datasets are imbalanced. Experiments show both the strengths and the limitations of the methods explored, providing direction for more refined and precise segmentation algorithms.

In summary, this chapter presents how reinforced active learning can enhance semantic segmentation by maintaining a balance between exploration and exploitation in reinforcement learning, leading to more effective and efficient image analysis in a wide range of applications.

5.2 Background

5.2.1 Semantic Segmentation

Semantic segmentation is a fundamental task in computer vision that assigns a class label to each pixel in an image, enabling a detailed understanding of visual content. This pixel-wise classification supports numerous applications, including autonomous driving, medical imaging, and robotics [83]. Unlike image classification, which assigns a single label to an entire image, semantic segmentation provides fine-grained spatial information that delineates object boundaries and spatial relationships within scenes.

However, semantic segmentation faces two major challenges: data imbalance and annotation inefficiency. In many real-world datasets, underrepresented classes lead

to biased model predictions that compromise performance for minority categories [84]. For example, clinical datasets may contain an abundance of images from certain demographic groups while lacking diversity from others, and natural image datasets often include dominant object categories that overshadow rarer ones [85]. Such imbalances cause models to overfit prevalent classes, reducing their ability to generalize across diverse data distributions [86].

Another challenge lies in obtaining high-quality pixel-wise annotations. Producing dense labels is both time-consuming and labor-intensive, often requiring expert annotators to mark every image at pixel-level precision [87]. This process is prone to human error and becomes impractical for large-scale datasets. As highlighted in [88], semantic segmentation datasets demand meticulous annotations to achieve accuracy and generalization, yet these datasets often remain constrained by limited labeling budgets. Hence, efficient annotation strategies such as active learning have become essential to minimize manual effort while maintaining annotation quality.

5.2.2 Active learning

Active learning is a machine learning paradigm that minimizes labeled data by selectively annotating the most informative samples. At each iteration, the model identifies a subset of samples from the unlabeled dataset that, if labeled, would most improve performance compared to random selection. This targeted annotation process enhances efficiency by focusing only on critical data points [89].

Active learning strategies are generally grouped into three types. The first, Membership Query Synthesis, involves the model generating new, informative samples

for training. The second, Stream-Based Selective Sampling, allows the model to decide whether to label each incoming sample based on its informativeness. The third, Pool-Based Sampling, selects the most informative examples from a large pool of unlabeled data.

Despite its advantages, active learning for semantic segmentation has received less attention than for image classification, mainly due to the higher annotation cost and complexity involved in pixel-level labeling, where each image requires detailed per-pixel annotation.

5.2.3 Reinforced Active Learning

Reinforced active learning integrates active learning with reinforcement learning to optimize sample selection policies through experience-based decision making. In this framework, a reinforcement learning agent learns to select data samples guided by a policy refined through iterative interaction with the learning environment. The agent receives feedback in the form of rewards, which correspond to measurable improvements in model performance after the inclusion of newly labeled samples [90]. This approach effectively balances exploration and exploitation, enabling the agent to explore diverse samples while prioritizing those expected to yield the highest performance gain. Reinforced active learning is particularly advantageous in low-data regimes and scenarios where annotation is expensive, as it maximizes model improvement under constrained labeling budgets [91].

5.2.4 Significance of Reinforced Active Learning in Semantic Segmentation

This work combines the benefits of reinforced active learning with semantic segmentation to handle dataset imbalances and annotation efficiency. The presented framework here capitalizes on state-of-the-art RL techniques in prioritizing the selection of informative and representative samples to alleviate the annotation burden and enhance the model over all classes, whether balanced or imbalanced.

In summary, the chapter has given a broad view of semantic segmentation, the challenges around it, and how active and reinforced learning methodologies could offer the potential to mitigate those challenges. The following sections will discuss the related work, proposed methodology, experimental setup, and results showing how effective the given framework of reinforced active learning is in improving semantic segmentation.

5.3 Related Work

Active learning is a methodology focused on achieving maximal performance gain with the smallest possible number of labeled samples. Its objective is to search within the unlabeled dataset for samples containing the greatest amount of information and present them to an oracle, such as human annotators, for labeling. This minimizes labeling cost while sustaining performance. Active learning approaches are generally classified into membership query synthesis [92, 93], stream-based selective sampling [94, 95], and pool-based strategies [96], depending on the application context [97]. Some methods combine multiple strategies to improve overall performance, as illustrated by

Shui et al. [98], who balanced sample diversity and uncertainty to enhance selection efficiency. Despite extensive study, active learning remains difficult to apply to high-dimensional data such as images, text, and video, so most research focuses on low-dimensional problems [99, 100].

Recent work has explored reinforcement learning to develop labeling policies that directly optimize active learning performance. Dhiman et al. [101] proposed a multimodal active learning framework using deep reinforcement learning to improve annotation efficiency in multimedia streaming applications. Gong et al. [102] introduced a meta-agent teaming active learning framework to reduce annotation labor in pose estimation. Sadigh et al. [103] formulated an active learning algorithm for inverse reinforcement learning based on human preferences between sample trajectories, while Kunapuli et al. [104] and Ezzeddine et al. [105] used expert feedback or preference elicitation to guide actions within a defined state space. These studies highlight how reinforcement learning can adaptively refine labeling strategies through learned policies rather than handcrafted heuristics.

Active learning has also been extended to semantic segmentation, where annotation costs are particularly high due to pixel-level labeling. Uncertainty-driven methods prioritize samples with high aleatoric uncertainty, as shown by Kampffmeyer et al. [106], while Jain et al. [107] emphasized diversity and representativeness through handcrafted metrics. Other works leveraged superpixel-based over-segmentation [108, 109] to reduce redundancy or applied domain-specific heuristics for biomedical images [110, 111]. Golestaneh et al. [112] further stressed the importance of self-consistency in selecting samples that remain invariant under simple transformations.

Following the emergence of the Deep Q-Network (DQN), various extensions were proposed to address its limitations and improve data efficiency. Double DQN [113, 114] reduces overestimation bias by separating action selection from evaluation. Prioritized experience replay [115] enhances learning efficiency by sampling informative transitions more frequently. The dueling network architecture [116] decouples state-value and advantage functions to improve generalization. Multi-step bootstrapping, as in A3C [117], balances bias and variance while accelerating reward propagation, and Noisy DQN [118] introduces stochastic layers to promote exploration. Building upon these improvements, our work presents a reinforced active learning agent that integrates these extensions to optimize sample selection for semantic segmentation tasks [88].

5.4 Methodology

5.4.1 Active Learning with Reinforcement Learning for Semantic Segmentation

Following Casanova et al. [119], we adopt their architecture to train the segmentation network and formulate the process as a Markov Decision Process (MDP). The active learning framework aims to enhance the performance of a segmentation network f , parameterized by θ , under a limited labeled sample budget B . At each iteration, a query network π , parameterized by ϕ , selects K regions from the unlabeled set U_t . These regions are then submitted to an oracle for labeling, augmenting the labeled set L_t . The segmentation network f is trained on the updated L_t , and its performance is evaluated using the Intersection-over-Union (IoU) metric. This process continues

until the labeling budget B is reached, enabling the model to achieve high-quality segmentation by efficiently selecting the most representative regions for annotation. The approach is data-centric, allowing the model to learn an acquisition strategy from previous active learning experiences.

The setup employs four distinct data partitions. The query network π is trained using a subset of labeled data D_T , iterating through the active learning process to learn an effective acquisition function that maximizes performance within the labeling budget B . The validation subset D_V is used to evaluate the query network. A separate subset D_R generates the reward signal by assessing the segmentation network’s performance, while another subset D_S , not larger than D_T , is used to construct the current state representation.

Formally, the MDP is defined as the tuple (S, A, r, T, γ) , where S denotes the set of states, A represents the actions composed of K sub-actions that depend on the segmentation network and labeled/unlabeled sets, $r : S \times A \rightarrow \mathbb{R}$ is the reward function based on improvement in mean IoU per class, $T : S \times A \times S \rightarrow \mathbb{R}$ is the state-transition function, and γ is the discount factor indicating the reduced value of future rewards compared to immediate ones.

An episode terminates when the labeling budget B is exhausted. At the end of each episode, the segmentation network f is reinitialized with weights θ_0 , and a new episode begins. The query policy π is trained by simulating multiple episodes, with weight updates performed after each time step by sampling transitions $(s_t, a_t, r_{t+1}, s_{t+1})$ from the experience replay buffer E . Figure 5.1 illustrates the overall training workflow.

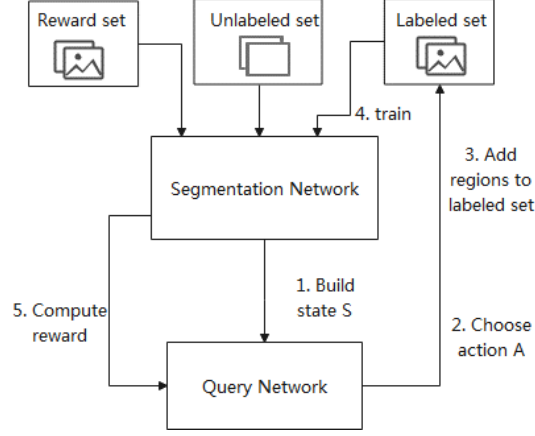


Fig. 5.1: The overall workflow of the active learning with Reinforcement learning in semantic segmentation.

5.4.2 Extensions to DQN

The evolution of Deep Q-Networks (DQN) led to several major extensions designed to overcome the limitations of the original framework and enhance its performance. These extensions expanded DQN’s capability, enabling more stable and efficient learning in complex environments. While Double DQN and Dueling DQN have been discussed in the previous chapter, the following section focuses on the remaining four extensions incorporated in our framework.

Prioritized experience replay: Schaul et al. [115] introduced the Prioritized Experience Replay mechanism, which improves sampling efficiency by assigning a priority value to each transition stored in memory. This prioritization allows the agent to replay more informative experiences more frequently, thereby accelerating learning and improving convergence stability.

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha} \quad (5.1)$$

Experiences are sampled based on their priorities, with α controlling the sampling bias. Priorities are determined by the temporal-difference (TD) error from the agent’s most recent update, directing more learning toward poorly predicted transitions and improving sample efficiency. Newly added transitions receive high priority, biasing replay toward recent data while retaining stochastic sampling to preserve exploration.

Emphasizing recent experience: Originally introduced to accelerate the convergence of Soft Actor-Critic (SAC) [120], this method can extend to various algorithms that benefit from emphasizing recent experiences. The key idea is to sample mini-batches from the replay buffer with increasing focus on recent data, ensuring faster adaptation. It follows two principles: (i) recent experiences are sampled more frequently, and (ii) older data are not allowed to overwrite newer ones. The Experience Replay Emphasis strategy thus enables efficient learning by prioritizing recent transitions while preserving knowledge from past policies through a simple yet effective sampling mechanism.

Adaptive epsilon greedy: The epsilon-greedy strategy balances exploration and exploitation by selecting a random action with probability ϵ and a greedy action with probability $1 - \epsilon$. For example, with $\epsilon = 0.3$, the agent explores randomly 30% of the time and exploits the best-known action 70% of the time. A refined version, Adaptive-epsilon-greedy [121], gradually decays ϵ from an initial value p_{init} (e.g., 0.6) to a final value p_{end} (e.g., 0.1) over n_{step} training epochs. This allows extensive exploration at the start and progressively shifts toward exploitation as training proceeds, following

the update rule:

$$r = \max \left(\frac{N - n_{\text{step}}}{N_0} \right) \quad (5.2)$$

$$\epsilon \leftarrow (p_{\text{init}} - p_{\text{end}}) \cdot r + p_{\text{end}} \quad (5.3)$$

This flexible scheme enables training to conclude with a low exploration probability p_{end} after n_{step} , allowing the agent to increasingly favor exploitation toward the end of training. A small residual exploration probability is maintained to ensure continued adaptability as the policy approaches convergence.

Noisy Network: Noisy networks are often preferred over the epsilon-greedy strategy as they enable more efficient and dynamic exploration during training. Instead of relying on random action selection, a Noisy Network [118] injects stochasticity directly into its parameters, producing smoother and more continuous exploration. It introduces a noisy linear layer that combines deterministic and stochastic components, allowing the network to learn both the optimal policy and the appropriate level of exploration adaptively.

$$y = (b + Wx) + (b_{\text{noisy}} \odot \epsilon_b + (W_{\text{noisy}} \odot \epsilon_w)x) \quad (5.4)$$

Where ϵ_w and ϵ_b are random variables and $\odot$ represents the Hadamard product. Over the long term, the network learns to ignore the noisy stream; however, it does so at different rates for different areas of the state space, allowing state-dependent exploration with a type of intrinsic self-annealing. More precisely, such a dynamic strategy of exploration can fine-tune the relationship between exploitation and exploration in

ways that favor adaptability and learning efficiency in complex settings.

Soft update for target network: The soft update target network is an important stabilization technique in deep reinforcement learning [122]. Unlike hard updates that periodically copy parameters from the main network to the target network, soft updates adjust the target parameters gradually toward the main network’s parameters. This smooth transition prevents abrupt changes that could destabilize learning. The update rate is controlled by τ , which determines the blending ratio. In the original DPG algorithm, τ was set to 0.001, and the target network update rule is defined as:

$$\theta_{\text{target}} = \theta \cdot \tau + \theta_{\text{target}} \cdot (1 - \tau) \quad (5.5)$$

The small value of τ allows the target network to gradually align with the Q-network, requiring frequent updates for effective adjustment. This approach enables the target network to track the main network’s changes smoothly, ensuring stable and consistent learning. It is especially beneficial in complex reinforcement learning tasks where stability during training is essential for achieving optimal performance.

5.5 Experimental Setup

This section describes the experimental setup used to evaluate the proposed reinforced active learning framework for semantic segmentation. It includes details on the datasets, data collection and preprocessing methods, evaluation metrics, comparative baselines, and the hardware and software configurations used in the experiments.

5.5.1 Dataset Description

The experiments use the CamVid dataset [123], containing 360×480 street scene images across 32 classes, as shown in Figure 5.2. Captured from a moving vehicle, the dataset provides diverse urban views. We focus on 11 main classes, where *Road*, *Building*, and *Sky* dominate (15.81%, 27.35%, and nearly all frames), while *Car* and *Pedestrian* occupy smaller regions (3.93% and 0.64%). Table 5.1 summarizes the evident class imbalance addressed in our study. The sequences include daytime and dusk scenes under varied lighting.

The dataset is split into 370 training, 104 validation, and 234 test images. From the training set, 100 labeled images form D_T , 10 images form D_S , and 260 images are used for baseline evaluation D_V . The validation set defines D_R , and the test set provides final segmentation results. Each image is divided into $K = 24$ regions of 80×90 pixels. We performed five runs with different random seeds, reporting the mean and standard deviation. Data augmentation includes horizontal flipping and random cropping to 224×224 .

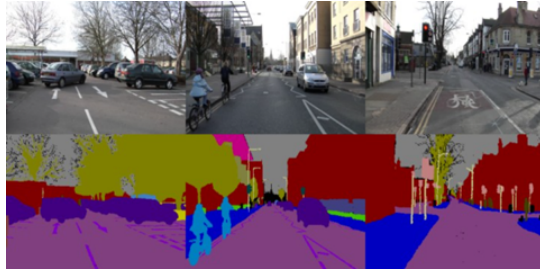


Fig. 5.2: Labeled frames from the video at 1Hz.

Fig. 5.1: Statistics for each class used in this study.

Class name	Percentage (%)	Occurrence
Road	27.3	701
Building	22.7	687
Sky	15.8	699
Tree	10.4	636
Sidewalk	6.33	672
Car	3.4	643
Column_Pole	0.98	698
Fence	1.43	363
Pedestrian	0.64	640
Bicyclist	0.53	365
Sign_Symbol	0.12	416

% shows the ratio of pixels.

Occurrence shows the number of occurrences over all images.

In addition to CamVid, the Grand Theft Auto V (GTA) dataset [124] was used for pre-training. This synthetic dataset provides pixel-level semantic annotations generated from the open-world game Grand Theft Auto V, offering diverse urban scenes from a vehicle’s perspective and richly annotated data for robust model initialization.

5.5.2 Data Collection and Preprocessing

The CamVid dataset was captured using a digital film camera under fixed conditions without auto-zoom, focus, or adjustments during the collection process. The cameras

were set to a fixed gain and shutter speed, and the aperture was maximized to prevent overexposure of white objects.

Preprocessing steps included:

- **Annotation:** Each image in the dataset was manually annotated to assign pixel-level semantic labels.
- **Data Augmentation:** To enhance the robustness of the model, data augmentation techniques such as horizontal flips and random crops (224×224 pixels) were applied to the images.
- **Region Division:** Each image was divided into K regions ($K = 24$) with a resolution of 80×90 pixels for the active learning process.

5.5.3 Evaluation Metrics

The active learning agent was trained on D_T using approximately 0.5k regions to optimize region selection for data-scarce conditions. Evaluation was performed on D_V under varying fixed budgets, where the agent accessed an increasing number of images. Once the budget was reached, the segmentation network was trained on L_T until meeting the early stopping criterion on D_R . The segmentation network f for all methods was pre-trained using the GTA dataset and D_T . Final performance was assessed on the CamVid test set using the Intersection over Union (IoU) metric.

5.5.4 Hardware and Software Configurations

The experiments were conducted using a single NVIDIA RTX A5000 GPU with 24 GB of VRAM. The following software configurations were used:

- Operating System: Ubuntu 20.04 LTS
- Deep Learning Framework: PyTorch
- Reinforcement Learning Libraries: OpenAI Gym, Stable Baselines3

Training the active learning agents required approximately 18 hours for 5 runs, while training the segmentation models to evaluate the active learning algorithms took a total of 8 hours for 5 runs at each of the 6 budgets.

5.6 Results

This section presents the experimental results of the proposed reinforced active learning framework for semantic segmentation. We compare multiple active learning techniques, highlight the improvements achieved by our method, and discuss their implications.

Figures 5.3 and 5.4 show performance across increasing budgets of labeled 128×128 pixel regions. The x-axis labeled “Budget” denotes the number of additional regions (in thousands) and the percentage of used unlabeled data. Each plot reports the mean and standard deviation over five runs. All segmentation networks were pre-trained on the GTA dataset and portions of their target datasets. The dashed line represents 96% of the best Intersection over Union (IoU) performance obtained when trained

with full supervision. As the preceding work achieves higher performance than other baselines, it is used as the reference model for subsequent comparisons.

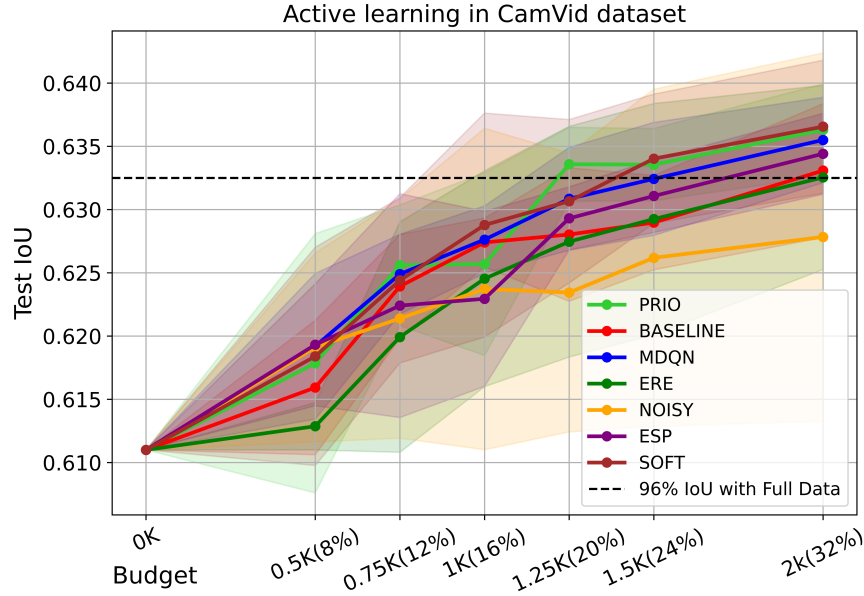


Fig. 5.3: Comparisons of various active learning methods.

Figure 5.3 compares the performance of several methods, including Prioritized Experience Replay (PRIO) [115], the reproduced DQN baseline (BASELINE) [119], Dueling Deep Q-Network (MDQN) [116], Emphasizing Recent Experiences (ERE) [120], Noisy Network (NOISY) [118], Adaptive Epsilon-Greedy (ESP) [121], and Soft Update for Target Network (SOFT) [122]. At 1.5k regions, several methods exceed 96% of the fully supervised model performance, demonstrating the efficiency of selective labeling. Among them, PRIO and SOFT consistently outperform other approaches, including the baseline, across all budgets except the 1k case for PRIO. The NOISY model performs the weakest, indicating that additional labeled data contribute little improvement. These results show that reinforced active learning

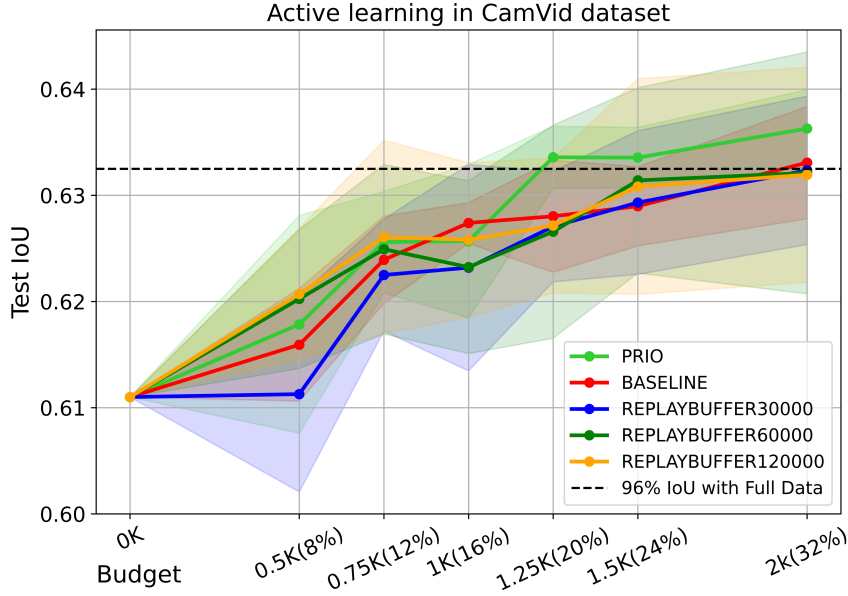


Fig. 5.4: Compare PRIO method with varying replay buffer sizes.

strategies, particularly those prioritizing or stabilizing updates, enhance segmentation performance by avoiding local minima and improving generalization.

Figure 5.4 compares the baseline and PRIO methods under different replay buffer sizes: 600, 30,000, 60,000, and 120,000. Performance remains stable for buffer sizes of 60,000 and 120,000. Notably, PRIO achieves superior results even with a much smaller buffer of 600, outperforming larger configurations by a significant margin, demonstrating its efficiency in utilizing limited experience memory.

The experimental results indicate that incorporating a soft target network update improves performance by allowing the target network to track the main network’s changes more smoothly, resulting in stable learning and faster convergence to an optimal policy. This stability yields more reliable Q-value estimates and better overall performance. The improvement observed with Dueling DQN arises from its separation

of value and advantage functions within Q-values. By decoupling these components, the model more effectively evaluates the importance of actions across different states, enhancing generalization and learning efficiency. This separation also reduces variance in the learned action-values, stabilizing training and mitigating Q-value overestimation or underestimation.

Prioritized Experience Replay (PER) enhances reinforcement learning by improving sample efficiency, stabilizing training, and promoting effective exploration. By assigning higher sampling probabilities to transitions with larger temporal-difference (TD) errors, PER accelerates convergence and facilitates efficient learning. Emphasizing rare but informative experiences enables the agent to handle critical situations more effectively. This prioritization ensures stable policy updates and faster learning, while focusing on transitions with high learning potential enhances decision-making. PER also reduces the bias of uniform sampling and mitigates high variance, leading to more accurate and stable Q-value updates and overall performance gains across reinforcement learning tasks.

Our experiments show that certain techniques, such as using Noisy Networks intended to promote diverse segmentation strategies, can degrade performance by introducing instability and reducing accuracy in specific scenarios. In complex environments with sparse rewards or high-dimensional state spaces, adjusting the exploration rate alone may not guarantee effective exploration. Poor tuning of exploration parameters or neglecting learning dynamics can disrupt the exploration–exploitation balance, even with adaptive epsilon-greedy strategies. Overemphasizing recent experiences in a Deep Q-Network (DQN) may further impair performance by lowering sample

efficiency, reducing generalization, increasing variance, and destabilizing training, hindering convergence to an optimal policy. Excessive focus on recent transitions also limits state-space exploration, preventing the discovery of rare yet critical states. Therefore, maintaining a balance between emphasizing recent experiences and preserving sample diversity is crucial for achieving stable and effective learning across the entire state space.

5.7 Conclusions

This chapter advances semantic segmentation by demonstrating the effectiveness of integrating active learning with reinforcement learning. The proposed framework addresses key challenges of dataset imbalance and annotation inefficiency, offering a practical and scalable solution for improving segmentation performance across diverse domains. By coupling reinforcement learning-based decision-making with active data selection, the framework enables models to focus on the most informative samples, reducing annotation costs while enhancing segmentation accuracy. These findings contribute to the ongoing development of more adaptive and data-efficient segmentation algorithms, reinforcing progress in both computer vision and machine learning.

In conclusion, reinforced active learning proves to be a promising strategy for overcoming the limitations of conventional annotation-driven approaches. It not only strengthens the foundation for semantic segmentation under constrained data conditions but also paves the way for future innovations that blend learning efficiency

with model generalization.

The next chapter extends this progression to zero-shot medical image segmentation, where large vision-language models (LVLMs) and foundation models such as the Segment Anything Model (SAM) enable segmentation without task-specific training or fine-tuning. This transition explores how prototype-guided methods using LVLM-generated spatial priors can further reduce annotation cost and improve cross-domain generalization.

Chapter 6

Prototype-Guided Zero-Shot Medical Image Segmentation with Large Vision-Language Models

Zero-shot methods represent a fundamental departure from traditional supervised learning, as they require no task-specific training or fine tuning. Instead, they leverage large vision-language models (LVLMs) and foundation models such as the Segment Anything Model (SAM) to generate segmentations directly, guided by bounding boxes or prompts. By replacing dependency on annotated datasets with visual prototypes and model-generated spatial priors, zero-shot segmentation addresses the dual challenges of annotation scarcity and domain shift. In this chapter, we introduce a prototype-guided framework that combines LVLM bounding box priors with SAM and CLIP embeddings, demonstrating its ability to achieve accurate segmentation across modalities without training or fine tuning.

6.1 Introduction

Accurate segmentation of anatomical structures is a critical step in medical image analysis, impacting downstream applications such as diagnosis, treatment planning, and disease monitoring. Traditional segmentation approaches rely on large annotated datasets and supervised training, which is impractical in many clinical domains due to annotation cost, domain variability, and privacy constraints [125, 126]. Moreover, even well-trained models often fail to generalize across different institutions, scanner vendors, and patient demographics due to domain shifts, leading to degraded performance and

the need for costly retraining.

Zero-shot segmentation addresses these challenges by enabling model deployment without any task-specific fine-tuning, allowing immediate application to new imaging modalities and anatomical targets. This dramatically reduces the burden on clinicians and facilitates scalable solutions in settings with limited annotation resources, such as rural hospitals or underrepresented populations. By leveraging foundation models like the Segment Anything Model (SAM) and large vision–language models (LVLMs), zero-shot methods offer a promising path toward training-free, adaptable, and equitable medical AI systems [127, 128, 129]. Robustness under data scarcity and domain variability makes zero-shot segmentation particularly well-suited for clinical translation.

Recent advances in foundation vision models, such as SAM [130] and CLIP [131], have shown impressive generalization on natural image datasets. However, their direct application to medical imaging is hindered by domain-specific differences in texture, contrast, and anatomy. The SaLIP framework [132] addressed this gap by integrating CLIP-based text–image similarity into the SAM segmentation pipeline, using enriched anatomical descriptions to guide mask selection. While effective, SaLIP [132] relies on language-based supervision, which introduces bias and reduces robustness when text descriptions are ambiguous or poorly aligned with domain-specific anatomy.

This paper proposes a new framework, termed LVLM Prototype Segmentation, that eliminates language dependence by integrating prototype based visual similarity and LVLM generated bounding boxes into the SAM segmentation pipeline. Specifically, our approach leverages structured prompt engineering to extract accurate anatomical bounding boxes from Gemini Pro 2.5 and uses class specific visual prototypes

(rather than text) for selecting the correct segmentation masks from the SAM output candidates.

Our contributions are as follows:

1. A prompt engineering strategy for improving LVLM-based bounding box estimation across diverse medical imaging modalities.
2. A prototype-based similarity ranking method that mitigates language bias and enhances segmentation accuracy, especially in anatomically ambiguous regions.
3. Comprehensive zero shot segmentation evaluation on three public datasets including brain MRI, fetal ultrasound, and chest X-ray, performed without any fine tuning or task specific training
4. Public release of code, bounding box annotations, and prompt templates to facilitate reproducibility and future work.

By combining LVLM-generated spatial priors with prototype-based visual retrieval, our framework outperforms prior zero-shot methods such as SaLIP [132] and approaches the performance of ground-truth-assisted methods, providing a robust and training-free path forward for medical image segmentation.

6.2 Background and Related Work

The Segment Anything Model (SAM) [130] has emerged as a versatile, prompt-driven segmentation framework capable of producing masks in a zero-shot manner. Its ability to segment without retraining has sparked interest in medical imaging,

where annotation is costly. However, direct application to medical data often yields suboptimal results due to domain gaps in appearance and structure boundaries [133, 134]. To address this, recent work has focused on test-time prompting strategies and semantic guidance without fine-tuning.

SaLIP [132] exemplifies this direction by combining SAM with CLIP [131] in a cascade. SAM first generates candidate masks, CLIP ranks them using text prompts enriched with GPT-3.5 descriptions of the target anatomy, and the top match is re-segmented with SAM for refinement. This approach improves zero-shot segmentation across modalities without retraining but relies on text-based similarity, which can introduce language bias and ambiguity when anatomical terms are imprecise or underrepresented in CLIP’s training data.

Other approaches automate prompt generation to guide SAM. SimSAM [135] simulates extensive point prompts and aggregates resulting masks to refine boundaries, while EviPrompt [136] places prompts in high-uncertainty regions using evidential reasoning. These methods improve mask quality without retraining but still depend on effective prompt placement and often require multiple SAM inferences per image.

Parallel work leverages prototype-based segmentation for generalization. Bian et al. [137] transferred learned “relation prototypes” between modalities to achieve cross-domain zero-shot segmentation, while Liu et al. [138] used CLIP-derived textual prototypes to unify organ and tumor segmentation in a single universal model. Such approaches demonstrate the utility of visual prototypes for removing language dependence but have not been combined with SAM’s mask proposal capabilities.

Current zero-shot medical segmentation methods either (1) depend heavily on

text-based CLIP similarity (e.g., SaLIP [132]), risking bias and poor coverage for rare terms, or (2) focus on prompt engineering without integrating strong semantic priors from LVLM model. Prototype-based methods mitigate language bias but lack the flexible mask generation and refinement pipeline of SAM. Additionally, existing works rarely exploit LVLMs capabilities (e.g., Gemini Vision) for precise bounding box estimation as priors to guide SAM prompting. The proposed framework addresses this gap by replacing textual prompts with *visual prototypes* for selecting the correct segmentation and leveraging LVLM-generated bounding boxes to drive *targeted point prompting*, improving both localization and segmentation accuracy in a fully zero-shot setting.

6.3 Methodology

6.3.1 Overview of the Proposed Framework

The proposed pipeline begins with prompt-engineered Gemini Pro to estimate precise bounding boxes for the target anatomical structure. From these bounding boxes, SAM is used to generate segmentation candidates by point prompting within the predicted region, while an additional candidate is generated directly from the initial Gemini-derived bounding box. Candidate selection is performed using CLIP, which computes similarity scores against a prototype embedding derived from multiple representative segmentation masks. The whole process is depicted in the Figure 6.1.

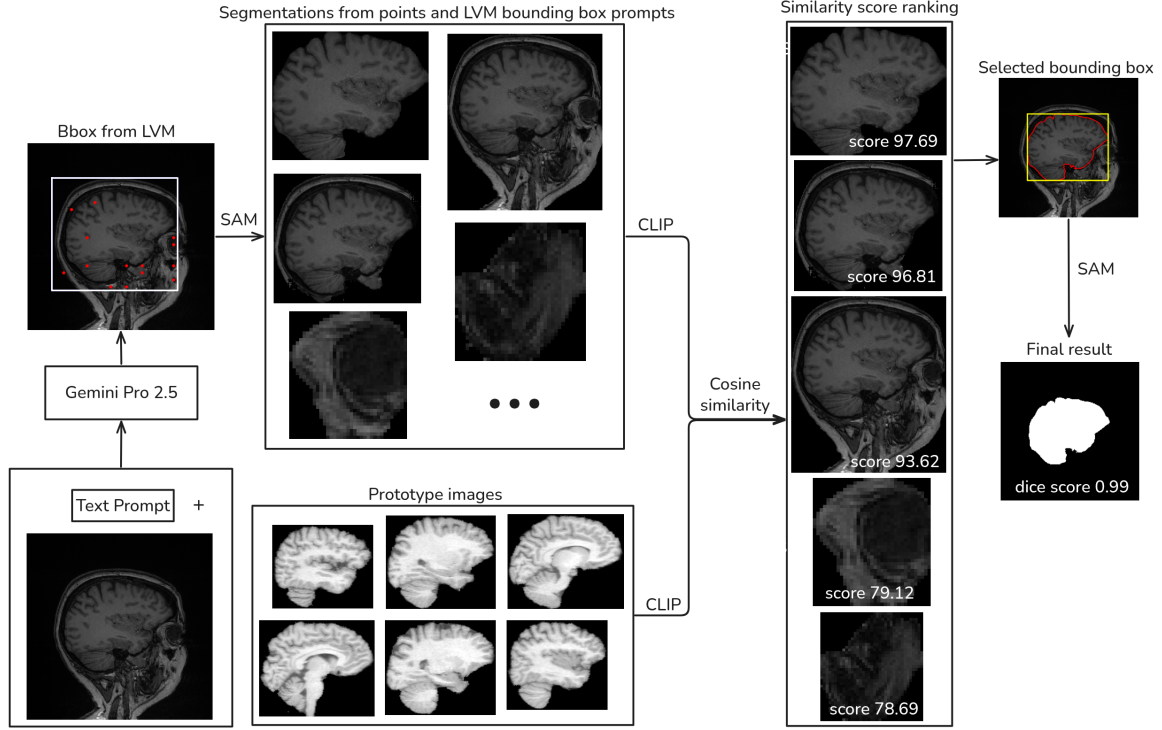


Fig. 6.1: Illustration of segmentation results using the proposed framework. The pipeline begins with an input image and a corresponding text prompt, which are packaged and sent to Gemini Pro 2.5 via its API. The model returns bounding box coordinates indicating the estimated target region. Within this region of interest (ROI), a set of evenly spaced points, along with the bounding box, are used as prompts for the SAM model to generate candidate segmentations. Each candidate is embedded using CLIP. Separately, prototype embeddings are computed by averaging embeddings from similar ROIs across various datasets. These prototypes are then compared to the candidate embeddings to compute similarity scores. The segmentations are ranked based on these scores, and the top-ranked result is selected. Finally, a refined bounding box derived from this selection is used to generate the final segmentation via SAM.

6.3.2 Bounding Box Estimation with Prompt-Engineered Large Vision-Language Models

Bounding box priors were generated using the Gemini Pro 2.5 API. To support large-scale annotation at a manageable cost, we implemented a batch inference pipeline, enabling the generation of approximately 10,000 bounding boxes in a single run. A systematic prompt design strategy was employed to ensure tight and anatomically relevant bounding boxes while minimizing the inclusion of irrelevant structures. Each input image was paired with a customized prompt and submitted to various vision-language models (VLMs), which returned bounding box coordinates representing the region of interest. Multiple prompt formulations were explored, incorporating modality-specific cues and spatial constraints. Certain designs improved box precision and reduced false positives, while others degraded performance or introduced ambiguity.

As of August 25 2025, GPT-4o demonstrated performance comparable to Gemini Pro 2.5; however, we selected Gemini Pro 2.5 for downstream use due to its consistent accuracy and available API funding. It also outperformed LLaMA-based models, as shown in Figure 6.3. Model outputs were returned in JSON format and integrated directly into our segmentation pipeline. While the bounding box estimations generated by Gemini across multiple datasets were not sufficiently precise to serve as final segmentations, they consistently localized the region of interest. Despite occasional boundary inaccuracies, these estimations provided reliable spatial priors that effectively guided the SAM model toward more accurate segmentations, as illustrated in Figure 6.2.

To further refine bounding box accuracy, we examined prompt engineering strate-

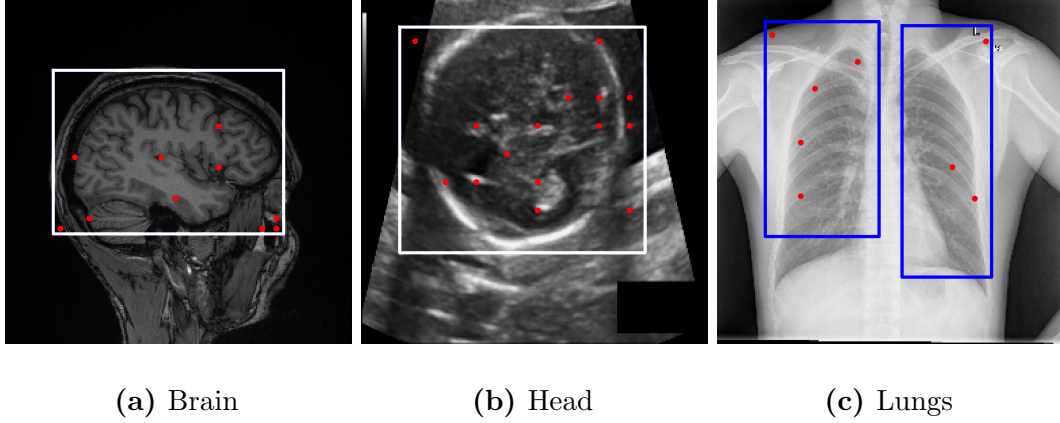


Fig. 6.2: Bounding box estimations by Gemini Pro 2.5 across different datasets.

gies. Three key elements were found to be critical: (1) a coordinate-setting variable (DIMENSIONS) that dynamically adjusts to image size, with the origin fixed at the top-left corner; (2) a precise anatomical description of the target region; and (3) a clearly specified output format. Full prompt templates are detailed in Appendix A.

6.3.3 Integration of LVLM Bounding Boxes with SAM and Prototype-Based Retrieval

We employed SAM ViT-H [130] as the base segmentation model, using bounding boxes produced by the vision-language model (LVLM) to localize candidate regions of interest. Each bounding box was resized to match SAM’s input resolution, after which SAM generated multiple candidate masks per box using both the LVLM-derived bounding box prompt and a set of evenly distributed point prompts within the box.

To identify the correct segmentation, we replaced CLIP-based text prompts with a prototype-based similarity retrieval strategy. Representative prototype images for each target class, drawn from different dataset cohorts to support zero shot

segmentation, were encoded using CLIP to compute an average class embedding. Each candidate image crop was also encoded with CLIP, and cosine similarity was computed between these crop embeddings and the corresponding class prototype embedding. The candidate with the highest similarity score was selected as the final segmentation, enabling robust anatomical alignment while reducing reliance on text prompt engineering as used in [132]. To make this process explicit, we formalize the computation of prototype embeddings and the retrieval of the most semantically similar crop in the shared embedding space as follows.

Let $\mathcal{C} = \{c_1, c_2, \dots, c_K\}$ be the set of target classes, and let $\{I_1, I_2, \dots, I_N\}$ denote the set of candidate crop images extracted from the input. For each class $c_k \in \mathcal{C}$, we define a set of M_k prototype (reference) images $\{R_1^{(k)}, R_2^{(k)}, \dots, R_{M_k}^{(k)}\}$.

Let $f(\cdot)$ denote the image encoder (e.g., CLIP), which maps an image to a d -dimensional embedding space. The embedding of each prototype image is computed as:

$$\mathbf{e}_m^{(k)} = \frac{f(R_m^{(k)})}{\|f(R_m^{(k)})\|}, \quad \text{for } m = 1, \dots, M_k, \quad (6.1)$$

The class prototype embedding $\mathbf{p}^{(k)}$ is obtained by averaging the normalized embeddings:

$$\mathbf{p}^{(k)} = \frac{1}{M_k} \sum_{m=1}^{M_k} \mathbf{e}_m^{(k)}, \quad (6.2)$$

$$\mathbf{p}^{(k)} \leftarrow \frac{\mathbf{p}^{(k)}}{\|\mathbf{p}^{(k)}\|}. \quad (6.3)$$

Each candidate crop I_n is encoded and normalized as:

$$\mathbf{z}_n = \frac{f(I_n)}{\|f(I_n)\|}, \quad \text{for } n = 1, \dots, N. \quad (6.4)$$

The cosine similarity between each crop embedding $\mathbf{z}_n$ and the prototype embedding $\mathbf{p}^{(k)}$ is:

$$s_n^{(k)} = \mathbf{z}_n^\top \mathbf{p}^{(k)}. \quad (6.5)$$

The final selected crop $\hat{n}^{(k)}$ for class c_k is:

$$\hat{n}^{(k)} = \arg \max_n s_n^{(k)}. \quad (6.6)$$

This approach selects the crop most semantically aligned with the class prototype in the shared embedding space, enabling robust segmentation without dependence on text prompts for mask selection.

6.4 Experiments

6.4.1 Datasets and Evaluation Metrics

Evaluation Datasets

The framework was evaluated on three datasets: (1) Calgary–Campinas Brain MRI for skull stripping, (2) HC18 fetal ultrasound for head circumference estimation, and (3) a chest X-ray dataset for lung field segmentation. Performance metrics included Dice similarity coefficient (DSC) and mean Intersection over Union (mIoU). Experiments are conducted on three publicly available medical image segmentation datasets:

- **CC359 Brain MRI** [139]: T1-weighted brain MRI scans from multiple scanners (GE 1.5T, GE 3T, Philips 1.5T, Philips 3T, Siemens 1.5T).
- **HC18 Fetal Head Ultrasound** [140]: 999 ultrasound images with manual fetal head annotations.

- **Chest X-ray Masks and Labels (CXRMAL)** [141]: 704 chest radiographs with lung masks.

Reference Prototype Sets

To evaluate the impact of different prototype sources on retrieval-based segmentation performance, we conducted controlled experiments using three distinct types of prototypes. For each dataset, only one prototype source was used at a time to isolate its contribution to the final segmentation results:

1. **Internal subsets (few-shot):** A small portion of each evaluation dataset was held out and used exclusively as prototypes. These samples were strictly excluded from the evaluation set to simulate a few-shot setting.
2. **External datasets (zero-shot):** A limited number of representative samples were selected from publicly available datasets of the same imaging modality but derived from different cohorts, enabling a zero-shot prototype setup.
3. **Anatomical diagrams (zero-shot):** Educational illustrations and schematic anatomical diagrams were used as prototypes. These were manually adjusted via geometric transformations such as flipping, rotation, or separation of left/right anatomical structures to better align with the spatial orientation of the evaluation data.

Brain MRI (CC359). Experiments were conducted separately using three types of prototypes: internal CC359 samples, Subject S01 from the LPBA40 dataset (T1-

weighted sagittal views) [142], and schematic brain diagrams, which were flipped when necessary to serve as diagram-based prototypes.

Fetal Head Ultrasound (HC18). Independent runs were performed using prototypes from internal HC18 samples or from a small set of annotated cases in a large-scale fetal head biometry dataset [143]. No suitable anatomical diagrams were available to serve as prototypes.

Lung Experiments were conducted using prototypes from internal data, external chest imaging samples of the same modality from the Darwin dataset [144], or stylized lung diagrams modified by separating the left and right lungs to match evaluation orientations.

By testing each prototype source in isolation, we were able to directly compare the performance impact of internal, external, and anatomical diagrams as prototypes. This design disentangles their contributions and highlights the strengths and limitations of each prototype strategy for zero-shot segmentation.

Evaluation Metrics

All images are first normalized to the range $[0, 255]$ and converted to three channels if originally grayscale. For volumetric modalities such as brain MRI and ultrasound, processing is performed on a slice-by-slice basis. Bounding boxes are expressed in pixel coordinates $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$ relative to each slice. Dice Similarity Coefficient

(DSC) and mean Intersection-over-Union (mIoU) are computed for each task:

$$\text{Dice} = \frac{2|P \cap G|}{|P| + |G|}, \quad (6.7)$$

$$\text{IoU} = \frac{|P \cap G|}{|P \cup G|}. \quad (6.8)$$

where P and G denote predicted and ground truth masks.

6.4.2 Performance

The Large Vision-Language Model bounding box priors improve localization, while prototype based similarity ranking refines mask selection, particularly in challenging scenarios involving low contrast, partial occlusion, or anatomical ambiguity. The proposed Gemini CLIP SAM framework achieves substantial improvements over baseline methods that rely solely on SAM point or text prompts.

Table 6.1 summarizes the quantitative results across all datasets, comparing our zero-shot framework with U-Net, SaLIP [132], GT-SAM (using ground-truth bounding boxes), and unprompted SAM (automatic segmentation of the entire image). To ensure consistency, we report both the original scores from SaLIP (as published) and rerun results obtained using the same prompts. We found that some previously reported scores could not be fully replicated, highlighting the challenges of prompt reproducibility and reinforcing the need for more robust alternatives such as prototype based retrieval.

Our method demonstrates consistent gains in both DSC and mIoU across all imaging modalities. For lung segmentation, DSC improves from 0.82 with SaLIP reruns to 0.90, and mIoU from 0.74 to 0.83. In fetal head segmentation, performance

risers from DSC 0.77 to 0.82 and mIoU from 0.68 to 0.74. Brain MRI benefits the most, with DSC improvements ranging from 0.02 to 0.14 depending on the scanner type. For example, Philips 1.5T improves from 0.80 to 0.94 in DSC and from 0.72 to 0.89 in mIoU. Siemens 3T scans achieve the highest overall performance, with our method reaching DSC 0.95 and mIoU 0.91 using zero shot prototype based similarity guided by LVLM priors.

In several cases, our method closely matches GT-SAM accuracy in Table 6.1. For example, both methods achieve DSC 0.93 and mIoU 0.88 on Philips 3T, while our approach attains DSC 0.94 and mIoU 0.90 on GE 3T compared with GT-SAM’s 0.96 and 0.93. Similarly, on Siemens 3T, our method reaches DSC 0.95 and mIoU 0.91, approaching GT-SAM (0.96 DSC, 0.90 mIoU). These results demonstrate the effectiveness of combining LVLM-generated bounding boxes with prototype-based similarity in narrowing the gap to an oracle bounding box setup.

Although U-Net achieves the highest performance in several brain MRI subsets (e.g., DSC 0.98 on GE 1.5T) in Figure 6.1, it is important to note that U-Net was fully supervised and trained on the target domain, whereas our method operates in a purely zero shot setting without any retraining or fine tuning. This highlights the strength of our approach in scenarios where annotated training data is limited or unavailable.

Overall, the integration of LVLM priors and prototype based ranking not only improves over text based baselines such as SaLIP [132] but also achieves segmentation quality comparable to that of GT-SAM, demonstrating strong generalization without the need for domain specific training.

Fig. 6.1: Performance comparison (DSC and mIoU) across models for different datasets. Best values in the Zero-shot and Few-shot categories are highlighted in bold.

ROI	Dataset	Training - Ground truth				Zero-shot						Few-shots	
		U-Net		GT-SAM		Un-prompted SAM		SaLIP (Text rerun)		Ours		Ours	
		DSC	mIoU	DSC	mIoU	DSC	mIoU	DSC	mIoU	DSC	mIoU	DSC	mIoU
Brain	GE 1.5	0.98	0.93	0.95	0.91	0.33	0.29	0.67	0.55	0.93	0.88	0.93	0.88
	GE 3	-	-	0.96	0.93	-	-	0.79	0.72	0.94	0.89	0.94	0.90
	Philips 1.5T	0.97	0.95	0.96	0.93	0.41	0.31	0.80	0.72	0.94	0.89	0.94	0.90
	Philips 3T	0.95	0.92	0.93	0.89	0.40	0.39	0.81	0.73	0.93	0.88	0.93	0.88
	Siemens 1.5T	0.97	0.95	0.95	0.91	0.39	0.26	0.68	0.57	0.91	0.86	0.92	0.86
	Siemens 3T	0.98	0.92	0.96	0.90	0.41	0.32	0.81	0.74	0.95	0.91	0.96	0.92
Lungs	X-ray	0.98	0.95	0.94	0.90	0.47	0.31	0.82	0.74	0.90	0.83	0.91	0.84
Fetal head	Ultrasound	0.95	0.91	0.95	0.91	0.55	0.40	0.77	0.68	0.82	0.74	0.86	0.78

6.4.3 Prototype-Based Similarity and LVLM Bounding Box

Table 6.2 compares CLIP similarity based on text prompts, as used in SaLIP [132], with our prototype-based similarity retrieval integrated with a large vision-language model (LVLM). Our approach consistently outperforms the text-based method across all datasets. The LVLM expands the candidate segmentation pool by incorporating outputs from bounding box prompts alongside point-based ones, while the prototype-based retrieval mechanism more effectively selects the most accurate segmentation. These quantitative improvements are shown in Table 6.2.

Qualitatively, the prototype-based method yields clearer and more anatomically accurate results. In brain MRI, it enhances delineation of cortical boundaries and ventricles while reducing false positives in non-brain regions. For lung segmentation, it enables better separation of the left and right lungs with reduced background leakage.

In fetal head ultrasound, it produces more complete and smoother head contours, especially under difficult imaging conditions. These qualitative gains align with the performance improvements in Table 6.2.

Moreover, incorporating a few labeled examples from the target dataset significantly improves performance across a range of modalities. In scenarios where no closely related dataset is available, simple anatomical diagrams representing the segmentation target can serve as effective prototypes. Although this results in a slight reduction in accuracy, the performance remains competitive, as demonstrated in Table 6.1.

As shown in Figure B.1, the text-based model frequently selects overly inclusive masks (e.g., covering most of the head in brain MRI), while the prototype-based approach more reliably identifies the correct segmentation. Other figures B.2 and B.3 provide representative examples from chest X-ray lung segmentation (CXRMAL), and fetal head ultrasound (HC18), highlighting the qualitative differences between our method, the SaLIP baseline, and the ground truth. The performance gains stem from both an improved pool of segmentation candidates and a more effective selection mechanism based on prototype similarity.

6.4.4 Ablations

To better understand the contribution of individual components in our method, we conducted a series of ablation studies. These experiments explore several design choices, including the role of prototype similarity in separating left and right lung regions, the effect of constraining point prompts to LVLM-derived bounding boxes, and the impact of prompt engineering and model selection for bounding box estimation.

Fig. 6.2: Performance comparison between text-based and prototype-based similarity retrieval with and without LVLM-generated bounding boxes (LVLM Bbox). Metrics are reported as Dice (DSC) and mIoU. Best results are in **bold**.

Methods		SaLIP				LVLM Bbox					
Similarity by		Reported)		Rerun		Text		Proto-zero-shot		Proto-diagram	
ROI	Dataset	DSC	mIoU	DSC	mIoU	DSC	mIoU	DSC	mIoU	DSC	mIoU
Brain	GE 1.5T	0.92	0.87	0.67	0.55	0.91	0.84	0.93	0.88	0.92	0.86
	GE 3T	-	-	0.79	0.72	0.91	0.86	0.94	0.89	0.91	0.84
	Philips 1.5T	0.94	0.85	0.80	0.72	0.90	0.84	0.94	0.89	0.92	0.87
	Philips 3T	0.89	0.80	0.81	0.73	0.89	0.83	0.93	0.88	0.92	0.86
	Siemens 1.5T	0.90	0.81	0.68	0.57	0.87	0.79	0.91	0.86	0.88	0.81
	Siemens 3T	0.93	0.85	0.81	0.74	0.93	0.89	0.95	0.91	0.93	0.88
Lungs	X-ray	0.83	0.76	0.82	0.74	0.90	0.83	0.90	0.83	0.90	0.83
Fetal head	Ultrasound	0.81	0.72	0.77	0.68	0.81	0.78	0.82	0.74	-	-

Through these targeted investigations, we aim to isolate the effects of each component and provide insight into how they contribute to the overall performance of the system.

Prototype Similarity for Separating Left and Right Lung

Previous work, such as SaLIP [132], has shown that text-based similarity alone is insufficient for distinguishing between left and right lung regions, largely because CLIP lacks spatial semantic understanding—specifically, it struggles to differentiate objects based on spatial alignment (i.e., left vs. right). In our experiments, we observed similar limitations: text-based CLIP similarity frequently resulted in overlapping or incorrect lung assignments, particularly in cases with ambiguous anatomical boundaries.

In contrast, our method leverages the LVLMs to explicitly estimate bounding boxes for the left and right lungs at the outset. This enables the formation of distinct candidate pools for each lung, allowing prototype-based similarity to more effectively retrieve the correct segmentation candidate. This approach significantly mitigates the ambiguity observed in purely text-based methods.

The quantitative results demonstrate the effectiveness of our approach. When integrated with text-based similarity, the LVLM achieves a Dice score of 0.90 ± 0.11 and an IoU of 0.83 ± 0.14 , surpassing the previously reported SaLIP baseline, which attained a Dice score of 0.82 and an IoU of 0.74 without incorporating the LVLM, as shown in Table 6.2. However, the performance gain from using our prototype-based model (Dice : 0.9001 ± 0.0825) was not substantially greater than that of the LVLM-enhanced text-based model (Dice : 0.8984), likely due to the high similarity among segmentation candidates generated from the same prompt applied to SAM.

Nonetheless, more substantial performance improvements with the prototype-based model are observed on other datasets, as illustrated in Figure B.3.

Restricting Point Prompts to LVLM Bounding Boxes

In the original SaLIP approach [132], point prompts were sampled uniformly across the entire image. In our modified version, point prompts were restricted to the region within the bounding box generated by the LVLM, with the aim of narrowing the search space to more relevant areas. However, this restriction did not result in a statistically significant improvement in Dice or mIoU. One possible explanation, particularly in the case of the lungs dataset, is that sampling point prompts across the entire image, as done in SaLIP, does not generate enough misleading segmentation candidates to confuse the final selection process. Nonetheless, this modification has shown more substantial performance gains on other datasets, as presented in Table 6.2.

Prompt Engineering and Choice of Large Vision-Language Models for Bounding Box Estimation

To derive reliable bounding box priors, we designed structured prompts that explicitly defined both the target anatomical structures and the spatial constraints of the bounding box. The final prompts used in our pipeline are provided in Appendix A. These prompts were tested across several LVLMs, including `maverick-llama-4`, `GPT-4V`, and Gemini Vision. While multiple models could generate bounding boxes, we selected Gemini Pro 2.5 for its consistent alignment with regions of interest and compatibility with available computational resources. In contrast, alternative models often produced

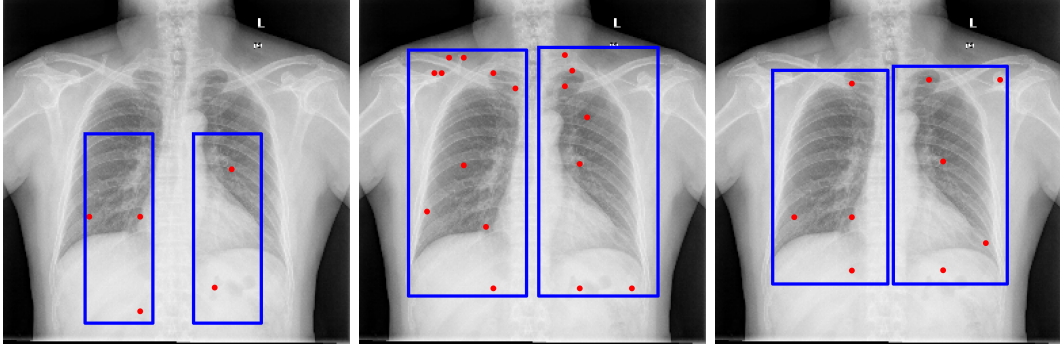


Fig. 6.3: Examples of lung bounding box estimations generated by different models: LLaMA-4-Maverick-17B-128E, GPT-4o, and Gemini Pro 2.5.

loose or misaligned boxes, which degraded downstream segmentation accuracy. Representative bounding box results from different models with the same prompt are shown in Figure 6.3, underscoring the importance of prompt engineering and the sensitivity of the pipeline to LVLM choice.

6.5 Discussion

This study introduces a prototype-guided zero-shot segmentation framework that leverages spatial priors generated by large vision-language models (LVLMs) and visual prototype-based retrieval to enhance mask selection within the Segment Anything Model (SAM) pipeline. The core innovation lies in decoupling mask selection from language embeddings by utilizing visual prototypes, and in replacing manually annotated bounding boxes with cost-efficient region proposals derived from LVLMs. Through systematic prompt engineering and the use of Gemini Pro 2.5, we show that anatomically relevant bounding boxes can be generated with high consistency across diverse imaging modalities. These bounding boxes act as spatial priors that improve

candidate generation within SAM and help constrain the region of interest, enabling improved localization without any task-specific training or fine-tuning.

Compared to previous approaches such as SaLIP [132], which rely on CLIP based text and image similarity for ranking segmentation candidates, our prototype retrieval strategy significantly improves robustness and generalization. Text prompts often struggle to capture spatial context or semantic precision, whereas our prototype embeddings, derived from representative image crops from external datasets or schematic anatomical diagrams, provide more discriminative representations that are less affected by prompt phrasing and linguistic ambiguity.

Quantitative results across three benchmark datasets (brain MRI, fetal head ultrasound, and chest X-ray) validate the effectiveness of our approach. In all cases, the method consistently outperforms zero-shot baselines and approaches the performance of oracle-guided setups (GT-SAM). The improvement is especially notable in fetal ultrasound, where domain shift and imaging variability commonly degrade the performance of conventional prompt-based methods. Furthermore, our framework offers a scalable and cost-effective solution: using Gemini Pro 2.5, we generated 10,000 bounding boxes in a single batch, demonstrating that LVLMs can serve as practical tools for semi-automated annotation pipelines.

Despite these advantages, several limitations remain. First, the quality of the bounding boxes depends on the choice of LVLm and the specificity of the prompt. While Gemini Pro 2.5 showed consistent performance, it is not infallible, and inaccurate bounding boxes may propagate errors into downstream segmentation. Second, the prototype retrieval component is sensitive to the quality of the prototype set. Non-

representative prototypes may reduce segmentation accuracy. Future work could explore automatic prototype selection, adaptive prototype updating, or integration with class-conditional diffusion models for prototype augmentation. Additionally, although we employ a general-purpose CLIP encoder to compute prototype similarity, domain-specific encoders trained on medical data (e.g., MedCLIP [145], BioCLIP [146]) may further improve performance. As LVLMS continue to evolve with increased accuracy, broader generalization, and reduced computational cost, our framework is well positioned to benefit. We envision improvements not only in performance but also in scalability across diverse clinical tasks. With continued advances in model architecture, training data, and inference efficiency, this approach may become a cornerstone of next-generation, training-free medical image analysis systems, enabling faster, more accessible, and more reliable deployment in real-world healthcare settings.

In summary, this work demonstrates that combining LVLM-generated priors with prototype-based mask selection can substantially improve the performance and reliability of zero-shot segmentation in medical imaging. By eliminating manual annotations, our framework provides a flexible and scalable solution that generalizes across imaging modalities and anatomical structures. The proposed method offers a promising direction for building robust, training-free medical image analysis systems suitable for rapid deployment in data-scarce or resource-constrained clinical environments.

6.6 Conclusions

This study introduces a novel framework for zero-shot medical image segmentation that integrates bounding box priors from a large vision-language model (Gemini Pro 2.5), prototype-guided mask retrieval using CLIP embeddings, and segmentation candidate generation via the Segment Anything Model (SAM). The proposed method removes the need for modality-specific fine tuning or task-specific annotations, enabling robust and scalable deployment across diverse imaging modalities, including brain MRI, fetal ultrasound, and chest X-ray.

By replacing language-based similarity with visual prototypes and leveraging spatial priors generated by the vision-language model, the framework addresses two key limitations of current prompt-based segmentation methods: language bias and unreliable localization. Extensive experiments demonstrate that the method consistently outperforms SaLIP [132] and unprompted SAM baselines, while narrowing the performance gap to fully supervised models. In several cases, its performance approaches or even matches that of oracle-guided segmentation using ground truth bounding boxes.

Importantly, the framework achieves strong results in a purely zero-shot setting, relying only on a small set of prototype images and general-purpose vision-language models. The use of structured prompt engineering for bounding box generation proved both accurate and cost efficient, enabling large-scale region proposals with minimal annotation effort. This capability paves the way for scalable semi-supervised labeling pipelines and broader accessibility of medical AI in low-resource settings.

The public release of code, bounding box annotations, and prompt templates enhances transparency and reproducibility, while facilitating integration into downstream segmentation pipelines and related research efforts.

In conclusion, combining large vision-language model-based localization with prototype-driven visual retrieval provides an effective strategy for training-free segmentation. This approach sets a new benchmark for zero-shot medical image analysis and highlights the broader evolution of segmentation methods—from traditional hand-crafted features to deep learning architectures, active learning strategies, and now zero-shot frameworks that redefine scalability and generalization in clinical practice.

Appendix A

Prompt engineering

Among all evaluated models, Gemini Pro 2.5 demonstrates the best performance for this task. All subsequent prompts are designed for this model.

A.1 Brain

Identity and Purpose

You are an expert radiologist specializing in brain anatomy. You will be provided with a sagittal medical image of a human brain. Your task is to identify and provide the coordinates for a bounding box that isolates the brain based on the specific anatomical landmarks defined below.

The origin for coordinates $(0, 0)$ is the top-left corner of the image. Assume the input image has dimensions of `DIMENSIONS` pixels.

Guidelines for Identifying the Bounding Box

- Top Edge ($y_{\min}$): The top edge of the bounding box must be horizontally aligned with the superior-most (highest) point of the frontal lobe.
- Right Edge ($x_{\max}$): The right edge must be vertically aligned with the anterior-most (rightmost) point of the frontal lobe.
- Left Edge ($x_{\min}$): The left edge must be vertically aligned with the posterior-most (leftmost) point of the occipital lobe.

- Bottom Edge ($y_{\max}$): The bottom edge must be horizontally aligned with the inferior-most (lowest) point of the cerebellum.

Output Instructions

The values must be in the following order:

$$[x_{\min}, y_{\min}, x_{\max}, y_{\max}]$$

Do not include any extra text, explanation, or formatting—only the list in square brackets.

Input

INPUT: [Insert the sagittal brain MRI image here]

A.2 Lungs

Identity and Purpose

You are an expert radiologist specializing in thoracic anatomy. You will be provided with coronal or axial medical images (e.g., CT scans). Your task is to identify and provide the tightest possible bounding box coordinates for the left lung and the right lung individually. The goal is to obtain an accurate estimation of the lung positions for later segmentation.

The origin of the x and y coordinates is the top-left corner of the image. Assume the input image has dimensions of **DIMENSIONS** pixels.

Guidelines for Identifying the Bounding Boxes

- Each bounding box must tightly enclose the entirety of its respective lung parenchyma.
- The boxes must strictly exclude the heart, mediastinum, great vessels, trachea, diaphragm, chest wall (ribs, muscle), and other non-lung tissues.
- Provide two distinct bounding boxes: one for the left lung and one for the right lung.
- Aim for the most compact bounding boxes possible to minimize empty space around the lung tissue.

Output Instructions

Provide the output as a single object with: left lung first, then right lung. Each bounding box must be represented as a list of four integers in the following order:

$$[x_{\min}, y_{\min}, x_{\max}, y_{\max}]$$

Do not include any extra text, explanation.

Example output format:

$$[[x_1, y_1, x_2, y_2], [x_3, y_3, x_4, y_4]]$$

Input

INPUT: [Insert the coronal or axial chest image here]

A.3 Fetal heads

Objective

As an expert obstetric sonographer, your task is to analyze a `DIMENSIONS` pixel axial ultrasound image of a fetal head and return its tightest possible bounding box coordinates. The coordinate origin $(0, 0)$ is the top-left corner of the image.

Bounding Box Criteria

- **Target Feature:** Identify the fetal head by its skull, which appears as a continuous, bright, elliptical boundary. This structure surrounds the darker brain tissue and is set against black amniotic fluid.
- **Enclosure:** The bounding box must completely enclose the entire fetal head, defined by the outer contour of the skull (calvarium).
- **Precise Fit:** The bounding box must be perfectly snug. All four edges of the box (top, bottom, left, and right) must touch the outer boundary of the fetal skull, ensuring there is no extra space between the box and the skull.

Output Instructions

Provide the output as a single list containing one list of four integers in the order:

$$[x_{\min}, y_{\min}, x_{\max}, y_{\max}]$$

Do not include any additional text, explanations, or labels.

Example output:

$$[[x_1, y_1, x_2, y_2]]$$

Input

INPUT: [Insert the axial fetal head ultrasound image here]

Appendix B

Visual results

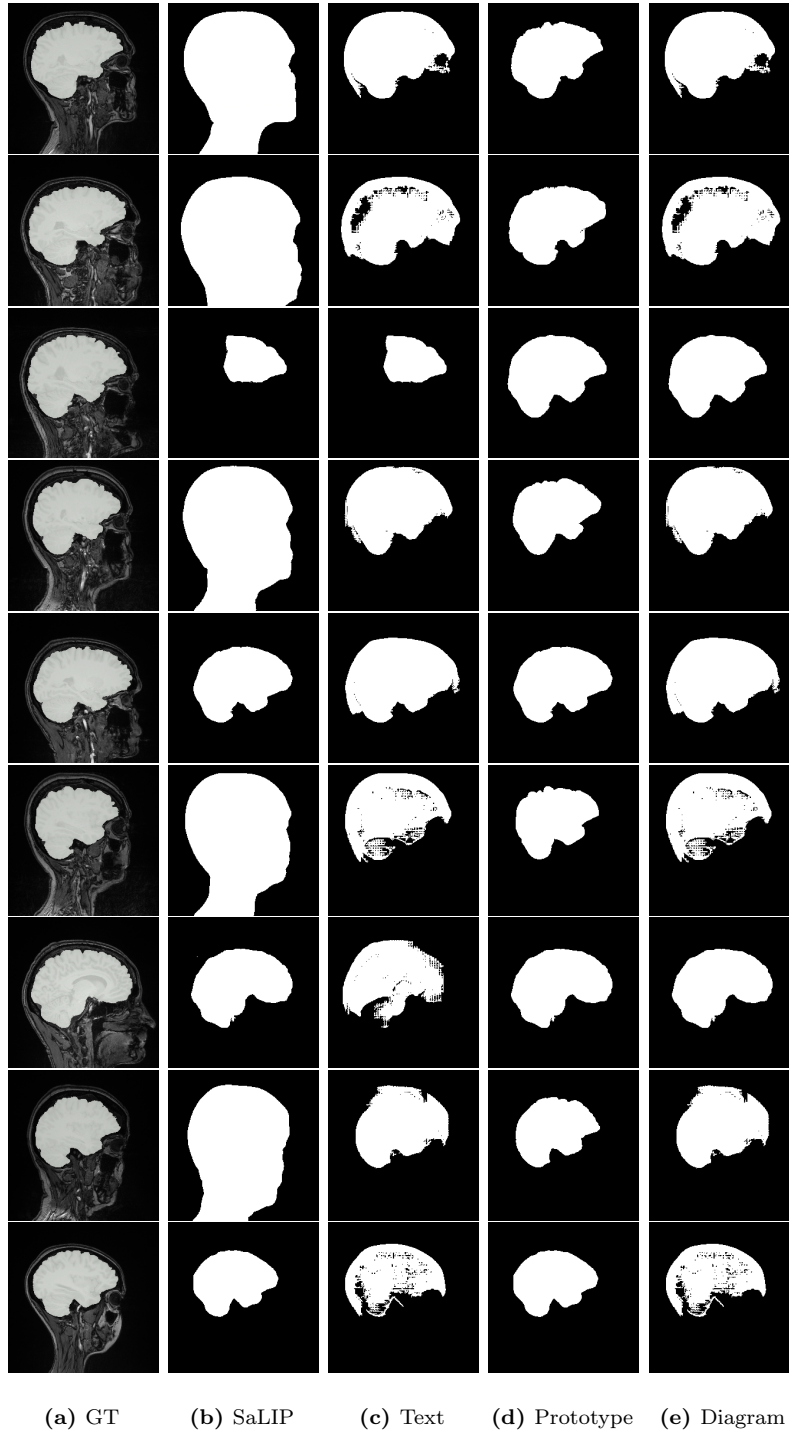


Fig. B.1: Brain MRI segmentation results. Left to right: (a) input with ground truth, (b) SaLIP, (c) LVLM with text retrieval, (d) LVLM with target-mask prototype, and (e) LVLM with anatomical-prototype guidance.

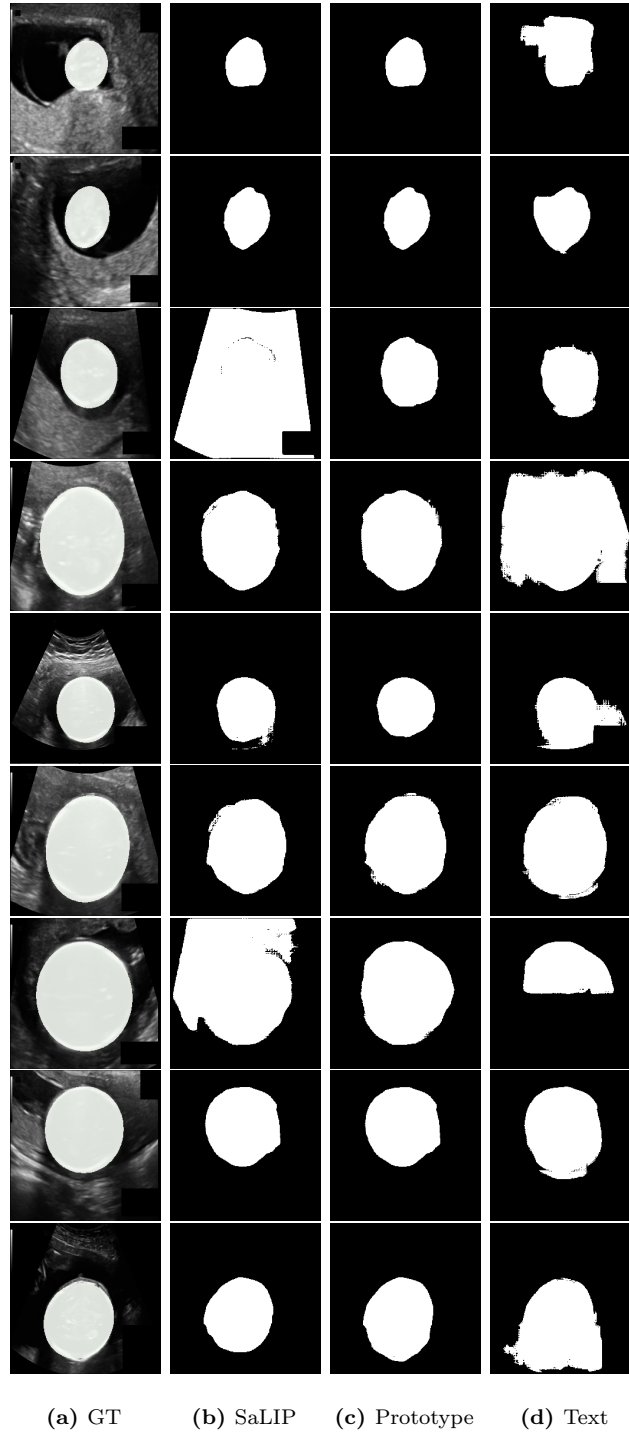


Fig. B.2: Fetal ultrasound segmentation results. Left to right: input, SaLIP, LVLM with mask prototype (ours), and LVLM with text-based retrieval.



Fig. B.3: Chest X-ray segmentation results. From left to right: (a) input with ground truth, (b) SaLIP, (c) LVLMM with text-based retrieval, (d) LVLMM with a target-mask prototype, and (e) LVLMM with an anatomical-illustration prototype.

Chapter 7

Conclusions

This dissertation has examined the progression of image segmentation methods, from traditional machine learning with handcrafted features to advanced deep learning architectures, active learning strategies, and zero-shot frameworks leveraging large vision-language models. Each chapter has highlighted both methodological progress and practical challenges, reflecting the broader trajectory of segmentation research across biomedical and environmental applications.

The early exploration of traditional machine learning demonstrated the effectiveness of pixel-wise classification using algorithms such as random forests and regression-based models. These approaches, applied within the Continuous Monitoring of Land Disturbance (COLD) framework, showed that even under limited data availability, meaningful segmentation and change detection could be achieved. However, their dependence on carefully designed handcrafted features and sensitivity to data imbalance limited their scalability and generalization across diverse scenarios.

The transition to deep learning, exemplified by the application of U-Net architectures for automatic rectum segmentation on low-field MRI, marked a significant step forward. U-Net’s ability to learn hierarchical spatial features directly from the data provided more robust and accurate segmentations, reducing inter-reader variability and supporting critical downstream tasks such as image registration for prostate biopsy guidance. Despite these advances, challenges remained in the form of small training datasets, annotation costs, and limited adaptability to new imaging modalities.

To address these issues, reinforced active learning was investigated as a strategy to improve efficiency and robustness in semantic segmentation. By integrating reinforcement learning with active learning, the framework selectively queried the most informative samples, reducing annotation costs while mitigating dataset imbalance. Experimental results highlighted its ability to improve segmentation performance under constrained labeling budgets, bridging the gap between supervised deep learning and scalable annotation pipelines. Yet, reliance on labeled data remained a persistent barrier for truly universal and training-free applications.

The final stage of this research introduced a prototype-guided zero-shot segmentation framework that combines bounding box priors from large vision-language models (Gemini Pro 2.5), candidate generation via the Segment Anything Model (SAM), and prototype-based similarity retrieval using CLIP embeddings. This method eliminates the need for modality-specific fine tuning or task-specific annotations, enabling robust segmentation across brain MRI, fetal ultrasound, and chest X-ray in a purely zero-shot setting. By replacing language-based similarity with visual prototypes and leveraging LVLm-generated spatial priors, the framework addressed key limitations of previous zero-shot approaches, narrowing the performance gap to fully supervised models. The results underscore the potential of zero-shot strategies as a new paradigm for medical image analysis, particularly in resource-limited clinical environments.

Taken together, the contributions of this dissertation illustrate the progressive shift in segmentation research: from handcrafted features that relied heavily on domain expertise, to deep learning architectures that learned directly from annotated data, to active learning frameworks that optimized limited labels, and ultimately to zero-shot

methods that dispense with task-specific training altogether. This trajectory reflects a broader movement in artificial intelligence toward generalization, adaptability, and reduced reliance on human supervision. As foundation models continue to advance in scale and capability, zero-shot approaches are poised to define the future of image segmentation by removing the constraints of exhaustive annotation and repetitive model retraining.

Future work should continue to explore ways of improving zero-shot segmentation, including refining bounding box estimation with domain-specific LVLMs, developing adaptive prototype selection strategies, and incorporating medical foundation encoders such as MedCLIP or BioCLIP. Beyond medical imaging, the frameworks developed here hold promise for environmental monitoring and other domains where annotation scarcity and domain variability pose significant challenges.

References

- [1] H. Pham, D. B. T. Le, M. Sadinski, R. Narayanan, A. Nacev, and B. Zheng, in *Medical Imaging 2023: Image-Guided Procedures, Robotic Interventions, and Modeling*, edited by C. A. Linte and J. H. Siewerdsen (2023), p. 72.
- [2] H. Pham, M. Jones, T. Gai, W. Islam, G. Danala, J. Jo, and B. Zheng, in *Medical Imaging 2022: Computer-Aided Diagnosis*, edited by K. M. Iftikharuddin, K. Drukker, M. A. Mazurowski, H. Lu, C. Muramatsu, and R. K. Samala (2022), p. 79.
- [3] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support* (2018), pp. 3–11.
- [4] O. Oktay *et al.*, arXiv preprint arXiv:1804.03999 (2018).
- [5] Z. Zhang, Q. Liu, and Y. Wang, *IEEE Geoscience and Remote Sensing Letters* **15**, 749 (2018).
- [6] O. Ronneberger, P. Fischer, and T. Brox, *CoRR* **abs/1505.04597**, (2015).
- [7] N. Ibtehaz and M. S. Rahman, *Neural Networks* **121**, 74 (2020).
- [8] A. Dosovitskiy *et al.*, in *International Conference on Learning Representations (ICLR)* (2021).
- [9] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, in *Advances in Neural Information Processing Systems (NeurIPS)* (2021).
- [10] H. Cao *et al.*, arXiv preprint arXiv:2105.05537 (2021).
- [11] B. Cheng, I. Misra, A. Schwing, A. Kirillov, and R. Girdhar, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 1290–1299.
- [12] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, in *International Conference on Machine Learning (ICML)* (2020), pp. 1597–1607.
- [13] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 9729–9738.
- [14] J.-B. Grill *et al.*, in *Advances in Neural Information Processing Systems (NeurIPS)* (2020).
- [15] S. Laine and T. Aila, in *International Conference on Learning Representations (ICLR)* (2017).
- [16] A. Tarvainen and H. Valpola, in *Advances in Neural Information Processing Systems (NeurIPS)* (2017).

- [17] A. Kirillov, E. Mintun, N. Ravi, H. Mao, G. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, in *arXiv preprint arXiv:2304.02643* (2023).
- [18] T. Lüddecke and A. S. Ecker, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 3479–3489.
- [19] Y. Wu *et al.*, arXiv preprint arXiv:2301.11798 (2023).
- [20] A. Hatamizadeh, V. Nath, Y. Tang, D. Yang, H. R. Roth, and D. Xu, in *Winter Conference on Applications of Computer Vision (WACV)* (2022).
- [21] Y. Ganin and V. Lempitsky, in *International Conference on Machine Learning (ICML)* (2015), pp. 1180–1189.
- [22] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, in *International Conference on Machine Learning (ICML)* (2018).
- [23] L. Hoyer, D. Dai, and L. Van Gool, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 9911–9921.
- [24] Y. Chen, Y. Zhou, X. Qiu, K. Cheng, F. Wang, and J. Tian, in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (2022).
- [25] Z. Zhu, J. Zhang, Z. Yang, A. H. Aljaddani, W. B. Cohen, S. Qiu, and C. Zhou, *Remote Sensing of Environment* **238**, 111116 (2020).
- [26] M. Melichar, K. Didan, A. Barreto-Muñoz, J. N. Duberstein, E. Jiménez Hernández, T. Crimmins, H. Li, M. Traphagen, K. A. Thomas, and P. L. Nagler, *Remote Sensing* **15**, 1266 (2023).
- [27] Z. Zhu and C. E. Woodcock, *Remote Sensing of Environment* **144**, 152 (2014).
- [28] B. Wang and Y. Yao, *Remote Sensing* **16**, 256 (2024).
- [29] R. Gharbia, *Journal of the Indian Society of Remote Sensing* **51**, 1511 (2023).
- [30] X. Wang, X. Xiao, Z. Zou, B. Chen, J. Ma, J. Dong, R. B. Doughty, Q. Zhong, Y. Qin, S. Dai, X. Li, B. Zhao, and B. Li, *Remote sensing of environment* **238**, 110987 (2020).
- [31] H. Cao, Y. Tian, Y. Liu, and R. Wang, *Scientific Reports* **14**, 16132 (2024).
- [32] M. Liu, J. Liu, and H. Hu, *Applied Sciences* **14**, 1344 (2024).
- [33] L. Yue, B. Li, S. Zhu, Q. Yuan, and H. Shen, *International Journal of Digital Earth* **16**, 210 (2023).
- [34] C. Huang, Y. Chen, S. Zhang, and J. Wu, *Reviews of Geophysics* **56**, 333 (2018).

- [35] S. Feng, S. Liu, G. Zhou, C. Gao, D. Sheng, W. Yan, Y. Wu, H. Gao, J. Jia, Z. Wang, Y. Ning, D. Ren, and M. Liu, *Journal of Hydrology: Regional Studies* **41**, 101111 (2022).
- [36] J. L. Dwyer, D. P. Roy, B. Sauer, C. B. Jenkerson, H. K. Zhang, and L. Lyburner, *Remote Sensing* **10**, 1363 (2018).
- [37] H. Xu, *International Journal of Remote Sensing* **27**, 3025 (2006).
- [38] N. Minh Trinh, *Journal of Hydro-meteorology* **8**, 1 (2024).
- [39] M. Mahua, F. Kasim, and N. Pasingi, *Jurnal Ilmu Kelautan SPERMONDE* **39** (2023).
- [40] S. K. Maryada, W. L. Booker, G. Danala, M. Jones, S. Mudduluru, H. Pham, D. F. Hougen, and B. Zheng, *An Efficient Synthetic Data Generation Algorithm to Improve Efficacy of Deep Learning Models of Medical Images*, 2022.
- [41] S. Sabour, N. Frosst, and G. E. Hinton, in *Advances in neural information processing systems* (2017), pp. 3856–3866.
- [42] L. Engstrom, D. Tsipras, L. Schmidt, and A. Madry, *arXiv preprint arXiv:1712.02779* **1**, 3 (2017).
- [43] M. A. Alcorn, Q. Li, Z. Gong, C. Wang, L. Mai, W.-S. Ku, and A. Nguyen, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 4845–4854.
- [44] G. E. Hinton, S. Sabour, and N. Frosst, in *International conference on learning representations* (2018).
- [45] G. E. Hinton, A. Krizhevsky, and S. D. Wang, in *International conference on artificial neural networks*, Springer (2011), pp. 44–51.
- [46] J. L. Ba, J. R. Kiros, and G. E. Hinton, *arXiv preprint arXiv:1607.06450* (2016).
- [47] Z. Zhao and S. Cheng, *Neural Networks* **143**, 690 (2021).
- [48] S. Yang, F. Lee, R. Miao, J. Cai, L. Chen, W. Yao, K. Kotani, and Q. Chen, *IEEE Access* **8**, 85007 (2020).
- [49] Y. Wu and K. He, in *Proceedings of the European conference on computer vision (ECCV)* (2018), pp. 3–19.
- [50] F. D. S. Ribeiro, K. Duarte, M. Everett, G. Leontidis, and M. Shah, *arXiv preprint arXiv:2206.02664* (2022).
- [51] A. A. T. Zade, M. J. Aziz, S. Masoudnia, A. Mirbagheri, and A. Ahmadian, *Computers in Biology and Medicine* **148**, 105917 (2022).
- [52] L. Wang, M. Tang, and X. Hu, *Scientific Reports* **13**, 3440 (2023).

- [53] Z. Zhang, S. Ye, P. Liao, Y. Liu, G. Su, and Y. Sun, in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, IEEE (2020), pp. 1544–1547.
- [54] A. A. Akinyelu, F. Zaccagna, J. T. Grist, M. Castelli, and L. Rundo, *Journal of imaging* **8**, 205 (2022).
- [55] M. Tran, V.-K. Vo-Ho, K. Quinn, H. Nguyen, K. Luu, and N. Le, *Deep Learning for Medical Image Analysis* (2024), pp. 75–97.
- [56] A. Avesta, Y. Hui, M. Aboian, J. Duncan, H. Krumholz, and S. Aneja, *American Journal of Neuroradiology* **44**, 562 (2023).
- [57] K. Bhagtani, A. K. S. Yadav, E. R. Bartusiak, Z. Xiang, R. Shao, S. Baireddy, and E. J. Delp, arXiv preprint arXiv:2204.12067 (2022).
- [58] H. H. Nguyen, J. Yamagishi, and I. Echizen, arXiv preprint arXiv:1910.12467 (2019).
- [59] A. Mehra, L. J. Spreeuwers, and N. Strisciuglio, in *VISIGRAPP* (2021).
- [60] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, in *Attention is All You Need* (2017).
- [61] J. Choi, H. Seo, S. Im, and M. Kang, in *Proceedings of the IEEE International Conference on Computer Vision Workshops* (2019), pp. 0–0.
- [62] W. Huang and F. Zhou, *Scientific Reports* **10**, 11383 (2020).
- [63] V. Mazzia, F. Salvetti, and M. Chiaberge, *Scientific reports* **11**, 14634 (2021).
- [64] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, (2017).
- [65] N. Xie and X. Wan, *IEEE Access* **9**, 129626 (2021).
- [66] Y.-H. H. Tsai, N. Srivastava, H. Goh, and R. Salakhutdinov, arXiv preprint arXiv:2002.04764 (2020).
- [67] F. D. S. Ribeiro, G. Leontidis, and S. D. Kollias, in *AAAI* (2020), pp. 3749–3756.
- [68] J. Gugglberger, D. Peer, and A. Rodríguez-Sánchez, arXiv preprint arXiv:2201.11091 (2022).
- [69] J. Hollósi, Á. Ballagi, and C. R. Pozna, *Algorithms* **16**, 336 (2023).
- [70] E. Alaoui-Elfels, T. Gadi, *et al.*, *International Journal of Intelligent Engineering & Systems* **16**, (2023).
- [71] N. Ohta, S. Kawai, and H. Nobuhara, *Journal of Advanced Computational Intelligence and Intelligent Informatics* **27**, 173 (2023).

- [72] J. Chen and Z. Liu, IEEE Transactions on Neural Networks and Learning Systems **31**, 2653 (2020).
- [73] K. T. Noor, A. Robles-Kelly, and B. Kusy, in *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, Springer (2022), pp. 163–172.
- [74] C. Pan and S. Velipasalar, in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 11996–12005.
- [75] A. Marchisio, A. Massa, V. Mrazek, B. Bussolino, M. Martina, and M. Shafique, in *Proceedings of the 39th International Conference on Computer-Aided Design* (2020), pp. 1–9.
- [76] G. Sun, S. Ding, T. Sun, C. Zhang, and W. Du, Applied Intelligence **52**, 3066 (2022).
- [77] Y. LeCun, F. J. Huang, and L. Bottou, in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, IEEE (2004), Vol. 2, pp. II–104.
- [78] H. Xiao, K. Rasul, and R. Vollgraf, arXiv preprint arXiv:1708.07747 (2017).
- [79] A. Krizhevsky, G. Hinton, *et al.*, (2009).
- [80] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, (2011).
- [81] H. Zhang, Z. Li, H. Zhao, Z. Li, and Y. Zhang, Applied Sciences **12**, 2634 (2022).
- [82] T. Nguyen, B.-S. Hua, and N. Le, CoRR **abs/2203.08965**, (2022).
- [83] Y. Huang, Z. Huang, and T. Jin, Applied Sciences **14**, 4724 (2024).
- [84] V. Ofitserov and A. Konushin, International Journal of Open Information Technologies **12**, 57 (2024).
- [85] S. V. Armstrong, S. Pallickara, S. Pallickara, S. Ghosh, J. F. Breidt, *et al.*, Journal of Machine Learning Research **12**, 123 (2024).
- [86] Y. Bi, Z. Chen, C. Liu, T. Liang, and F. Zheng, Machine Vision and Applications **35**, 74 (2024).
- [87] P. Varangaonkar and S. Rode, Multimedia Tools and Applications **80**, 1 (2024).
- [88] D. Han, P. Huong, and S. Cheng, Journal of Electronic & Information Systems **5**, 45 (2023).
- [89] H. Hajiabadi, C. Gerking, L. Hilbert, and A. Koziolk, Journal of Systems and Software **211**, 111986 (2024).

- [90] C. Yu, J. Zhu, and X. Li, arXiv preprint arXiv:2402.10074 **abs/2402.10074**, 1 (2024).
- [91] Y. Zhang, H. Tong, Y. Xia, Y. Zhu, Y. Chi, and L. Ying, in *Proceedings of the AAAI Conference on Artificial Intelligence* (2022), No. 8, pp. 9118–9126.
- [92] D. Angluin, *Machine learning* **2**, 319 (1988).
- [93] R. D. King, K. E. Whelan, F. M. Jones, P. G. Reiser, C. H. Bryant, S. H. Muggleton, D. B. Kell, and S. G. Oliver, *Nature* **427**, 247 (2004).
- [94] I. Dagan and S. P. Engelson, *Machine Learning Proceedings 1995* (1995), pp. 150–157.
- [95] V. Krishnamurthy, *IEEE Transactions on Signal Processing* **50**, 1382 (2002).
- [96] D. D. Lewis, in *ACM SIGIR Forum*, ACM (1995), No. 2, pp. 13–19.
- [97] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, *ACM computing surveys (CSUR)* **54**, 1 (2021).
- [98] C. Shui, F. Zhou, C. Gagné, and B. Wang, in *International Conference on Artificial Intelligence and Statistics*, PMLR (2020), pp. 1308–1318.
- [99] B. Settles, in *Active learning and experimental design workshop in conjunction with AISTATS 2010*, JMLR Workshop and Conference Proceedings (2011), pp. 1–18.
- [100] J. M. Hernández-Lobato and R. Adams, in *International Conference on Machine Learning*, PMLR (2015), pp. 1861–1869.
- [101] G. Dhiman, A. V. Kumar, R. Nirmalan, S. Sujitha, K. Srihari, N. Yuvaraj, P. Arulprakash, and R. A. Raja, *Multimedia Tools and Applications* **82**, 5343 (2023).
- [102] J. Gong, Z. Fan, Q. Ke, H. Rahmani, and J. Liu, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 11079–11089.
- [103] D. Sadigh, A. D. Dragan, S. Sastry, and S. A. Seshia, *Active preference-based learning of reward functions* (2017).
- [104] G. Kunapuli, P. Odom, J. W. Shavlik, and S. Natarajan, in *2013 IEEE 13th International Conference on Data Mining*, IEEE (2013), pp. 409–418.
- [105] A. Ezzeddine, N. Mourad, B. N. Araabi, and M. N. Ahmadabadi, *Expert Systems with Applications* **112**, 331 (2018).
- [106] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2016), pp. 1–9.

- [107] S. D. Jain and K. Grauman, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 2864–2873.
- [108] A. Vezhnevets, J. M. Buhmann, and V. Ferrari, in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE (2012), pp. 3162–3169.
- [109] K. Konyushkova, R. Sznitman, and P. Fua, in *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 2974–2982.
- [110] J. Aklilu and S. Yeung, in *Machine Learning for Healthcare Conference*, PMLR (2022), pp. 892–911.
- [111] X. Shu, Y. Yang, R. Xie, J. Liu, X. Chang, and B. Wu, *SSRN Electronic Journal* **2022**, 1 (2022).
- [112] S. A. Golestaneh and K. M. Kitani, arXiv preprint arXiv:2008.01860 **abs/2008.01860**, 1 (2020).
- [113] H. Van Hasselt, A. Guez, and D. Silver, in *Proceedings of the AAAI Conference on Artificial Intelligence* (2016), No. 1.
- [114] H. Hasselt, *Advances in neural information processing systems* **23**, 2613 (2010).
- [115] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, arXiv preprint arXiv:1511.05952 **abs/1511.05952**, 1 (2015).
- [116] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, in *International Conference on Machine Learning*, PMLR (2016), pp. 1995–2003.
- [117] M. Babaeizadeh, I. Frosio, S. Tyree, J. Clemons, and J. Kautz, arXiv preprint arXiv:1611.06256 **abs/1611.06256**, 1 (2016).
- [118] M. Fortunato, M. G. Azar, B. Piot, J. Menick, I. Osband, A. Graves, V. Mnih, R. Munos, D. Hassabis, O. Pietquin, C. Blundell, and S. Legg, *Noisy Networks for Exploration*, 2019.
- [119] A. Casanova, P. O. Pinheiro, N. Rostamzadeh, and C. J. Pal, arXiv preprint arXiv:2002.06583 **abs/2002.06583**, 1 (2020).
- [120] C. Wang and K. Ross, arXiv preprint arXiv:1906.04009 **abs/1906.04009**, 1 (2019).
- [121] M. Tokic and G. Palm, in *Annual Conference on Artificial Intelligence* (2011), pp. 335–346.
- [122] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, arXiv preprint arXiv:1509.02971 **abs/1509.02971**, 1 (2015).
- [123] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, in *Computer Vision—ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12–18, 2008, Proceedings, Part I*, Springer (2008), pp. 44–57.

- [124] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, in *European Conference on Computer Vision (ECCV)*, Vol. 9906 of *LNCS*, edited by B. Leibe, J. Matas, N. Sebe, and M. Welling (2016), pp. 102–118.
- [125] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. v. d. Laak, B. v. Ginneken, and C. I. Sánchez, *Medical Image Analysis* **42**, 60 (2017), arXiv:1702.05747 [cs].
- [126] O. Ronneberger, P. Fischer, and T. Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, 2015, arXiv:1505.04597 [cs].
- [127] K. Fan, L. Liang, H. Li, W. Situ, W. Zhao, and G. Li, *Bioengineering* **12**, 608 (2025).
- [128] Y. Xu, J. Tang, A. Men, and Q. Chen, arXiv preprint arXiv:2311.06400 (2023).
- [129] T. Zhou, Y. Zhang, Y. Zhou, Y. Wu, and C. Gong, arXiv preprint arXiv:2304.07583 (2023).
- [130] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer, SAM 2: Segment Anything in Images and Videos, 2024, arXiv:2408.00714 [cs].
- [131] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, Learning Transferable Visual Models From Natural Language Supervision, 2021, arXiv:2103.00020 [cs].
- [132] S. Aleem, F. Wang, M. Maniparambil, E. Arazo, J. Dietlmeier, G. Silvestre, K. Curran, N. E. O'Connor, and S. Little, Test-Time Adaptation with SaLIP: A Cascade of SAM and CLIP for Zero shot Medical Image Segmentation, 2024, arXiv:2404.06362.
- [133] K. Fan, L. Liang, H. Li, W. Situ, W. Zhao, and G. Li, *Bioengineering* **12**, 608 (2025).
- [134] T. Zhou, Y. Zhang, Y. Zhou, Y. Wu, and C. Gong, Can SAM Segment Polyps?, 2023, arXiv:2304.07583 [cs].
- [135] B. Towle, X. Chen, and K. Zhou, SimSAM: Zero-shot Medical Image Segmentation via Simulated Interaction, 2024, arXiv:2406.00663 [cs].
- [136] Y. Xu, J. Tang, A. Men, and Q. Chen, EviPrompt: A Training-Free Evidential Prompt Generation Method for Segment Anything Model in Medical Images, 2023, arXiv:2311.06400 [cs].
- [137] C. Bian, C. Yuan, K. Ma, S. Yu, D. Wei, and Y. Zheng, *IEEE Transactions on Medical Imaging* **41**, 1043 (2022).

- [138] J. Liu, Y. Zhang, J.-N. Chen, J. Xiao, Y. Lu, B. A. Landman, Y. Yuan, A. Yuille, Y. Tang, and Z. Zhou, in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)* (2023), pp. 21095–21107, arXiv:2301.00785 [eess].
- [139] R. Souza, O. Lucena, J. Garrafa, D. Gobbi, M. Saluzzi, S. Appenzeller, L. Rittner, R. Frayne, and R. Lotufo, *NeuroImage* **170**, 482 (2018).
- [140] T. L. A. van den Heuvel, D. de Bruijn, C. L. de Korte, and B. van Ginneken, *PLOS ONE* **13**, e0200412 (2018), hC18 Challenge Dataset, DOI: 10.5281/zenodo.1322001.
- [141] N. Sharma, *Diagnostics* **12**, 2132 (2022), chest X-ray Masks and Labels (CXR-MAL) Dataset, 704 chest radiographs with lung masks.
- [142] D. W. Shattuck, M. Mirza, V. Adisetiyo, C. Hojatkashani, G. Salamon, K. L. Narr, R. A. Poldrack, R. M. Bilder, and A. W. Toga, LPBA40 Subjects Native Space: MRI Data and Brain Masks in Native Space, Laboratory of Neuro Imaging (LONI), University of Southern California, 2007, https://www.loni.usc.edu/research/atlas_downloads.
- [143] M. Alzubaidi, M. Agus, M. Makhoul, F. Anver, K. Alyafei, and M. Househ, Large-Scale Annotation Dataset for Fetal Head Biometry in Ultrasound Images, Zenodo, 2023, dataset hosted on Zenodo under CC BY 4.0 license.
- [144] V. Danilov, A. Proutski, A. Kirpich, D. Litmanovich, and Y. Gankin, Chest X-ray dataset for lung segmentation, Mendeley Data, Version 2, 2022, available via Mendeley Data—Chest X-ray dataset for lung segmentation.
- [145] Z. Wang, Z. Wu, D. Agarwal, and J. Sun, MedCLIP: Contrastive Learning from Unpaired Medical Images and Text, 2022, arXiv:2210.10163 [cs].
- [146] S. Stevens, J. Wu, M. J. Thompson, E. G. Campolongo, C. H. Song, D. E. Carlyn, L. Dong, W. M. Dahdul, C. Stewart, T. Berger-Wolf, W.-L. Chao, and Y. Su, BioCLIP: A Vision Foundation Model for the Tree of Life, 2024, arXiv:2311.18803 [cs].