

**UC Davis**

**UC Davis Electronic Theses and Dissertations**

**Title**

Robust Machine Learning Directed Toward Pathology Imaging Applications with Limited Labeled Data

**Permalink**

<https://escholarship.org/uc/item/8dt4k1q4>

**Author**

Lai, Zhengfeng

**Publication Date**

2023

Peer reviewed|Thesis/dissertation

Robust Machine Learning Directed Toward Pathology Imaging  
Applications with Limited Labeled Data

By

ZHENG FENG LAI  
DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Electrical and Computer Engineering

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

---

Chen-Nee Chuah, Chair

---

Sen-Ching Cheung

---

Brittany N. Dugger

Committee in Charge

2023



Copyright © 2023 by  
ZHENG FENG LAI  
*All rights reserved.*

# CONTENTS

|                                                                              |           |
|------------------------------------------------------------------------------|-----------|
| Abstract . . . . .                                                           | vii       |
| Acknowledgments . . . . .                                                    | ix        |
| <b>1 Introduction</b>                                                        | <b>1</b>  |
| 1.1 Overview . . . . .                                                       | 1         |
| 1.2 Contributions . . . . .                                                  | 5         |
| 1.2.1 Supervised learning framework for pathology image analysis . . . . .   | 5         |
| 1.2.2 Enhanced semi-supervised learning for practical scenarios . . . . .    | 5         |
| 1.2.3 Unsupervised domain adaptation . . . . .                               | 8         |
| 1.2.4 Efficient adaptation of CLIP for data-efficient learning . . . . .     | 10        |
| 1.2.5 List of proposed technical methods . . . . .                           | 11        |
| 1.2.6 List of publications . . . . .                                         | 11        |
| <b>2 BrainSec: Supervised Learning Pipeline for Pathology Image Analysis</b> | <b>14</b> |
| 2.1 Related work . . . . .                                                   | 16        |
| 2.1.1 Image processing-based segmentation . . . . .                          | 16        |
| 2.1.2 Deep learning-based segmentation . . . . .                             | 16        |
| 2.1.3 Patch-based approach for ultra-high resolution images . . . . .        | 20        |
| 2.2 Datasets . . . . .                                                       | 21        |
| 2.2.1 30-WSI Dataset . . . . .                                               | 21        |
| 2.2.2 Training data preparation . . . . .                                    | 22        |
| 2.2.3 130-WSI and 52-Heteroarea Datasets . . . . .                           | 22        |
| 2.3 Methodology . . . . .                                                    | 23        |
| 2.3.1 BrainSec . . . . .                                                     | 24        |
| 2.3.2 BrainSec+ . . . . .                                                    | 27        |
| 2.3.3 Pathology quantification & visualization with BrainSec . . . . .       | 30        |
| 2.4 Experiments . . . . .                                                    | 33        |
| 2.4.1 Quantitative results . . . . .                                         | 33        |

|          |                                                                   |           |
|----------|-------------------------------------------------------------------|-----------|
| 2.4.2    | Training convergence of BrainSec . . . . .                        | 35        |
| 2.4.3    | Computation complexity . . . . .                                  | 37        |
| 2.4.4    | Segmentation visualization . . . . .                              | 37        |
| 2.4.5    | BrainSec vs. BrainSec+ . . . . .                                  | 38        |
| 2.4.6    | Pathology quantification with BrainSec . . . . .                  | 39        |
| 2.4.7    | Grad-CAM interpretation . . . . .                                 | 39        |
| 2.4.8    | Evaluation on broader neuroanatomical areas . . . . .             | 41        |
| 2.5      | Discussion . . . . .                                              | 42        |
| <b>3</b> | <b>Enhanced Semi-supervised Learning to Improve Self-training</b> | <b>44</b> |
| 3.1      | Methodology . . . . .                                             | 45        |
| 3.1.1    | First stage: self-supervised pre-training . . . . .               | 45        |
| 3.1.2    | Second stage: standard FixMatch . . . . .                         | 46        |
| 3.2      | Experiments . . . . .                                             | 47        |
| 3.2.1    | Data preparation . . . . .                                        | 47        |
| 3.2.2    | Effectiveness of the first stage . . . . .                        | 47        |
| 3.2.3    | Numerical results . . . . .                                       | 48        |
| 3.2.4    | Segmentation visualization . . . . .                              | 49        |
| 3.3      | Discussion . . . . .                                              | 50        |
| <b>4</b> | <b>Class-Imbalanced Semi-supervised Learning</b>                  | <b>51</b> |
| 4.1      | Related work . . . . .                                            | 53        |
| 4.2      | Methodology . . . . .                                             | 54        |
| 4.2.1    | Preliminaries . . . . .                                           | 54        |
| 4.2.2    | A closer look at the weight in consistency loss . . . . .         | 55        |
| 4.2.3    | SAW: Smoothed adaptive weighting scheme . . . . .                 | 57        |
| 4.2.4    | SaR: Self-adaptive refinement on soft labels . . . . .            | 58        |
| 4.3      | Experiments . . . . .                                             | 63        |
| 4.3.1    | CIFAR10-LT . . . . .                                              | 64        |
| 4.3.2    | CIFAR100-LT and STL-10 . . . . .                                  | 66        |

|          |                                                                                                                                     |           |
|----------|-------------------------------------------------------------------------------------------------------------------------------------|-----------|
| 4.3.3    | Empirical analysis . . . . .                                                                                                        | 67        |
| 4.4      | Discussion . . . . .                                                                                                                | 75        |
| <b>5</b> | <b>Joint Semi-supervised and Active Learning Framework</b>                                                                          | <b>76</b> |
| 5.1      | Related work . . . . .                                                                                                              | 77        |
| 5.2      | Methodology . . . . .                                                                                                               | 78        |
| 5.2.1    | Problem setup . . . . .                                                                                                             | 78        |
| 5.2.2    | Semi-supervised active learning . . . . .                                                                                           | 79        |
| 5.2.3    | Region-based selection criterion . . . . .                                                                                          | 81        |
| 5.3      | Preliminary results . . . . .                                                                                                       | 83        |
| 5.3.1    | Training setup . . . . .                                                                                                            | 83        |
| 5.3.2    | Quantitative comparison . . . . .                                                                                                   | 84        |
| 5.3.3    | Segmentation visualization . . . . .                                                                                                | 86        |
| 5.3.4    | A deeper look into AL cycles . . . . .                                                                                              | 87        |
| 5.3.5    | Analysis on number of regions selected . . . . .                                                                                    | 89        |
| 5.4      | Discussion . . . . .                                                                                                                | 89        |
| <b>6</b> | <b>A Deployable Semi-supervised Learning with Informative Active Annotation for Pathology Image Classification and Segmentation</b> | <b>91</b> |
| 6.1      | Introduction . . . . .                                                                                                              | 92        |
| 6.2      | Related work . . . . .                                                                                                              | 94        |
| 6.3      | Methodology . . . . .                                                                                                               | 95        |
| 6.3.1    | Class-aware Adaptive Pseudo Labeling (APL) . . . . .                                                                                | 96        |
| 6.3.2    | Informative Active Annotation (IAA): Deep fusion of SSL and AL . . .                                                                | 97        |
| 6.3.3    | Extension to gigapixel-level segmentation tasks . . . . .                                                                           | 100       |
| 6.4      | Experiments . . . . .                                                                                                               | 100       |
| 6.4.1    | Main results on pathology image classification and segmentation tasks .                                                             | 101       |
| 6.4.2    | Empirical analysis on Semi-Path . . . . .                                                                                           | 104       |
| 6.5      | Discussion . . . . .                                                                                                                | 108       |

## **7 PADCLIP: Pseudo-labeling with Adaptive Debiasing in CLIP for Unsupervised**

|                                                         |            |
|---------------------------------------------------------|------------|
| <b>Domain Adaptation</b>                                | <b>109</b> |
| 7.1 Introduction . . . . .                              | 110        |
| 7.2 Related works . . . . .                             | 112        |
| 7.3 Methodology . . . . .                               | 114        |
| 7.3.1 CLIP in UDA . . . . .                             | 114        |
| 7.3.2 Catastrophic Forgetting Measurement . . . . .     | 115        |
| 7.3.3 Pseudo-labeling and inter-class bias . . . . .    | 116        |
| 7.3.4 Pseudo-labeling with Adaptive Debiasing . . . . . | 117        |
| 7.3.5 End-to-end trainable pipeline . . . . .           | 120        |
| 7.4 Experimental setup . . . . .                        | 120        |
| 7.5 Results . . . . .                                   | 121        |
| 7.5.1 Ablation studies . . . . .                        | 121        |
| 7.5.2 External comparison . . . . .                     | 123        |
| 7.5.3 Generalization of our method . . . . .            | 124        |
| 7.5.4 Computational complexity . . . . .                | 125        |
| 7.6 Conclusion . . . . .                                | 126        |

## **8 Path-CLIP: Efficient Adaptation of CLIP for Pathology Image Analysis with Limited Data**

|                                                                             |            |
|-----------------------------------------------------------------------------|------------|
| <b>ited Data</b>                                                            | <b>128</b> |
| 8.1 Introduction . . . . .                                                  | 129        |
| 8.2 Related work . . . . .                                                  | 132        |
| 8.2.1 Language-vision model . . . . .                                       | 132        |
| 8.2.2 Language-vision training in medical domain . . . . .                  | 132        |
| 8.2.3 CLIP fine-tuning and adaptation . . . . .                             | 133        |
| 8.3 Methodology . . . . .                                                   | 133        |
| 8.3.1 Language-Vision Pre-training . . . . .                                | 134        |
| 8.3.2 Residual Feature Connection . . . . .                                 | 136        |
| 8.3.3 Self-adaptive residual ratio . . . . .                                | 139        |
| 8.3.4 Path-CLIP: end-to-end adaptation of CLIP to pathology tasks . . . . . | 140        |

|          |                                                                   |            |
|----------|-------------------------------------------------------------------|------------|
| 8.4      | Experiments . . . . .                                             | 141        |
| 8.4.1    | Datasets and setup . . . . .                                      | 141        |
| 8.4.2    | Main results . . . . .                                            | 144        |
| 8.4.3    | Ablation studies . . . . .                                        | 147        |
| 8.4.4    | Importance of alleviating overfitting . . . . .                   | 148        |
| 8.5      | Discussion . . . . .                                              | 149        |
| <b>9</b> | <b>Future Work</b>                                                | <b>152</b> |
| 9.1      | Heterogeneity in pathology image analysis tasks . . . . .         | 152        |
| 9.1.1    | Expanded datasets for robustness analysis . . . . .               | 152        |
| 9.1.2    | Extend SSL to more pathology image analysis tasks . . . . .       | 152        |
| 9.1.3    | Class-imbalanced and fairness issue in pathology images . . . . . | 153        |
| 9.2      | Transferability and semi-supervised transfer learning . . . . .   | 153        |
| 9.3      | From interpretability to continual learning . . . . .             | 155        |

## ABSTRACT

### **Robust Machine Learning Directed Toward Pathology Imaging Applications with Limited Labeled Data**

Neurodegenerative diseases are defined by a progressive loss of selectively vulnerable neuronal populations. Hence, augmenting the ability of a pathologist/expert for the detection/localization of specific pathologies with data-driven deep learning (DL) approaches could have a transformative impact. DL models trained on high-quality annotated datasets can provide a cost-effective and reliable means for deeper phenotyping of neurodegenerative and neurologic diseases. Leveraging digital slide scanners such as the Leica Aperio AT2 and Zeiss Axio Z1, physical tissue slides can be digitized into gigapixel digital whole slide images (WSIs) containing a rich set of information, including diverse morphological patterns, cellular features, and tissue architecture. However, there are severe challenges to fully realize the digitalization and automation of detecting these pathologies and neurodegenerative features. First, the tissue sections can be scanned at  $\times 20$  or  $\times 40$  objective magnification, resulting in ultra-high resolution WSIs that require a large amount of memory space for storage and processing. To alleviate this issue, we design a general patch-based approach and use plaque quantification in grey and white matter as one example. In this study case, our automated framework can classify, localize, quantify, and visualize the distribution of each type of plaque in grey and white matter, separately. We have tested it across different scanners, brain areas, and storage formats to benchmark its generalizability.

Second, although the proposed supervised learning pipelines can achieve promising results, their performance heavily relies on the quality and quantity of the data and annotations. However, manual annotation to build comprehensive and well-annotated digital pathology datasets can be time-consuming and labor-intensive, which is not scalable for diverse pathological tasks. Therefore, to relieve the heavy reliance on the labeled data, we investigate semi-supervised learning (SSL) and propose four SSL frameworks to enhance the applicability and deployment of SSL algorithms. For example, we propose an imbalanced SSL but make no assumption on the distribution of the unlabeled data; we improve pseudo-labeling and design a deep fusion of SSL and active learning to further reduce the labeling efforts.

Third, besides the annotation cost, data collection can also be expensive and challenging due to the scarcity of pathology data and the heterogeneity of pathology datasets, which can vary significantly regarding image resolution, staining techniques, and scanning equipment. Therefore, to reduce the data collection effort, we study unsupervised domain adaptation (UDA) and efficient adaptation of large-scale pre-trained vision-language models (VLMs). UDA proposes to reduce data annotation costs by leveraging a labeled source domain to transfer the knowledge into an unlabeled target domain. We propose a UDA pipeline to bridge the domain gap and make domain adaptation data and label-efficient. On the other hand, VLMs have shown promising results in their transferability across different downstream tasks. However, traditional fine-tuning may involve billions of parameters to be tuned, which is computation-expensive and has overfitting issues when the downstream task with limited data availability. We propose a parameter-efficient pipeline to adapt VLMs to multiple downstream tasks in a scalable way in terms of processing time and computational cost. In conclusion, our proposed pipelines can potentially achieve data and label-efficient learning for diverse pathological imaging tasks.



## ACKNOWLEDGMENTS

I would like to send my earnest gratitude to my major advisor, Professor Chen-Nee Chuah. Without Prof. Chuah's support and help along the way, none of these would be possible to be completed during my Ph.D. program. Throughout my graduate journey, Prof. Chuah has been an unwavering source of inspiration, guidance, and support. Her efforts have seamlessly connected me to invaluable collaboration and internship opportunities, significantly enriching my academic and professional growth. I extend my heartfelt gratitude to Prof. Chuah for standing by me through the highs and lows of this challenging and rewarding path.

I would like to sincerely thank Professor Sen-ching Cheung and Professor Brittany N. Dugger for mentoring, revising my manuscripts, and serving on my dissertation committee and for all the helpful guidance to my dissertation. I am deeply grateful for Prof. Cheung's expert guidance and astute editing of the computer vision chapters. Additionally, I extend my sincere thanks to Prof. Dugger for her invaluable guidance and meticulous edits on the pathology imaging chapters, which elevated their professional quality. I would also like to express my thanks to Professor Soheil Ghiasi and Professor Jinyi Qi for serving on my qualification exam committee.

I would like to thank all of Rubinet members for their great help. Lastly, I would like to thank my family for their constant support.

# Chapter 1

## Introduction

### 1.1 Overview

The aging population is increasing all over the world: the population of people aged 60 or older is over 900 million and expected to be over 1,800 million by 2050 [1]. Alzheimer disease (AD), one of the neurodegenerative diseases that increase risks with age, resulted in more than 120,000 deaths in 2018, and the number of patients may rise to 13.8 million in the U.S. by the middle of this century [2]. Thus, understanding the underlying pathological features behind this disease becomes a critical task. To study the progression of AD, neuropathologists visually inspect postmortem brain tissue slides and identify subtly differentiated hallmarks that are the gold standard for the diagnosis of AD [3]. Traditionally, neuropathologists compare these morphologies by studying brain tissue slides using microscopes. Recently, Whole-slide Imaging (WSI) has been an increasingly popular method for assessing brain tissues with the help of a scanner, which scans the tissues into whole slide images (WSIs) of ultra-high resolution to preserve minimal details [4]. Furthermore, WSIs can be viewed through computer software (such as Aperio ImageScope and QuPath [5]) so trained personnel can identify and quantify the morphologies with more computational approaches [6, 7].

There are many pathologies within the brain that define neurodegenerative diseases, such as AD, and their locations can be essential to gain insights into disease pathophysiology. For example, one hallmark pathological feature of AD is the presence of extracellular Amyloid- $\beta$

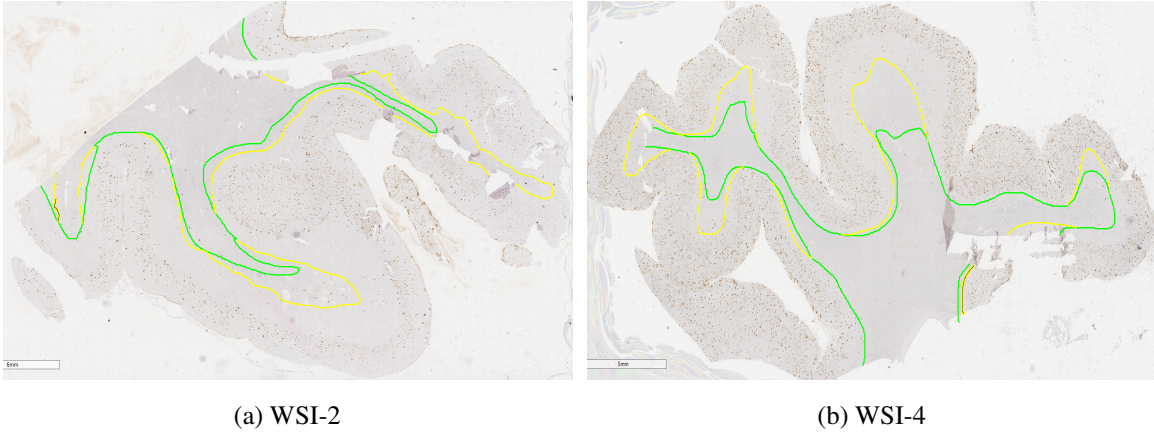


Figure 1.1. WSIs annotated by two trained personnel. (Green and yellow colors denote the two independent annotations.)

plaques in the human brain [8, 9]. Most plaques are found in grey matter (GM), while some have also been reported in white matter (WM) [10]. To study the distribution, there are three tasks: plaque detection, GM/WM region segmentation, and plaque quantification in these two regions. However, these manual assessments can be highly time-consuming and labor-extensive considering the wide existence of plaques in brain tissue and the gigapixel resolution of WSIs. Moreover, manual assessment could be subjective and have inter-rater reliability issues. For example, Figure 1.1 shows two cases of manually segmenting GM and WM, in each of which two trained personnel draw considerably different boundaries between GM and WM in the same brain WSIs. Therefore, to gain more insights into the progress of AD, it is necessary to construct an end-to-end system that can achieve the following goals with minimal manual efforts: 1) automatic detection and identification of pathological plaques; 2) segmentation of GM and WM regions; 3) quantification and visualization of the pathological plaques in GM and WM regions.

There are many proposed methods in both traditional image processing and deep learning for WSI segmentation. However, these have been applied to the tissue slides primarily from other parts of the human body (tongue, breast, lymph node, rectum [11], skin [12]), anatomical brain (instead of tissue level)[13]. Although these methods are helpful in the analysis of WSIs in various applications, several issues limit the direct applications of these methods to our WSIs for the analysis of brain tissues.

First, there are unwanted artifacts in WSIs, such as tissue residues, tissue folds, bubbles, and

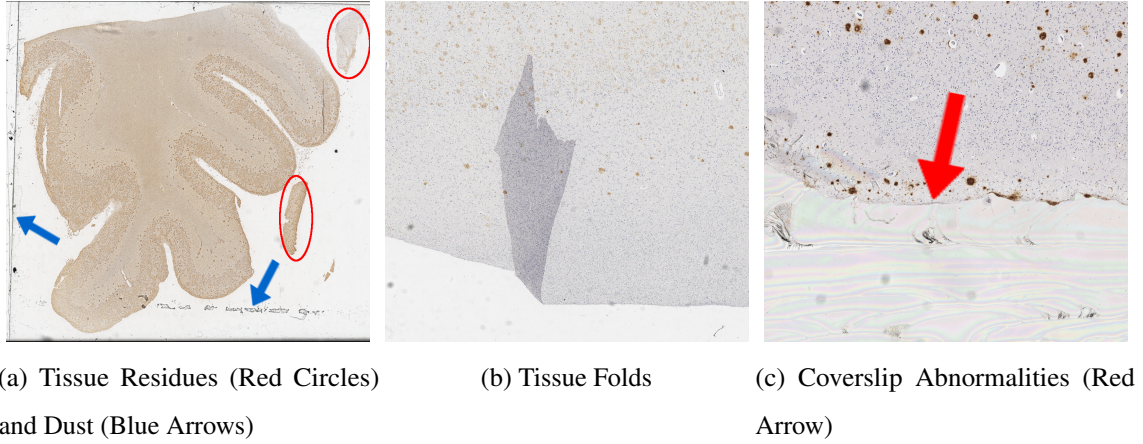


Figure 1.2. Sample Artifacts in WSIs.

dust. A sample of these artifacts is as shown in Figure 1.2. These artifacts can set obstacles for segmentation methods to separate the background accurately.

Second, our WSIs are scanned at an ultra-high resolution to retain cellular-level medical details (at 20x or 40x magnification). Thus, the average resolution of these WSIs exceeds 50,000 by 50,000 pixels, making the files very large and complex to process in a timely manner. To address this resolution issue, current methods can be divided into two categories: either downsample original images of ultra-high resolution or divide original images into small patches for separate processing [14]. To provide a proof of concept for a workflow for GM/WM segmentation algorithm and to minimize the need for manual segmentation and tracings, we utilize 30 WSIs with annotations, far less than previous datasets used in [14]. Besides, the minute plaques in our WSIs are pathological hallmarks of AD. Plaques and other cellular structures with sizes ranging from  $20 \times 20$  to  $100 \times 100$  in WSIs are distinguishable features of GM and WM, while simple downsampling may make them less apparent or even disappear [15], which results in their limited performance. We need to preserve the distributions and characteristics of plaques in WSIs to study AD. Therefore, in this study, we aim to focus on the patch-based approach instead of downsampling.

Third, due to the gigapixel resolution, the annotation efforts are substantial. The limited labeled dataset hinders the wide adoption of fully-supervised learning models. For example, a recent study [16] claimed it requires more than 30,000 labeled tiles from gigapixel WSIs to

achieve the well-defined performance of deep learning models, which requires labor-intensive labeling [14]. Furthermore, the labeling cost could be much higher when experts must do annotations (in this study, doctors are required for medical problems) [17]. Semi-supervised learning (SSL) can leverage unlabeled data to reduce the labeling efforts and has shown promising results in computer vision tasks [17, 18, 19, 20]. However, the efforts to deploy them on gigapixel pathology tasks remain limited [21, 22, 23, 24]. Therefore, we investigate SSL and propose deployable SSL frameworks for pathology image analysis.

SSL may still need to collect a large amount of unlabeled data. Building comprehensive and well-annotated digital pathology datasets can be challenging due to the scarcity of annotated data and heterogeneity of pathology datasets, which can vary significantly in terms of image resolution, staining techniques, and scanning equipment [25]. This heterogeneity poses a challenge in the generalized learning process. Specifically, limited data availability can hinder the intensive end-to-end training of deep networks, and fine-tuning the traditional models has overfitting issues, and a large domain gap between the pre-trained data and pathology images impacts the performance. Also, the requirement of multiple learning tasks for the complete pathology image analysis is time-consuming and can be resource-hungry. Hence, it is important to develop deep learning methods that can perform multiple learning tasks and efficiently learn the rich set of semantic information from WSIs. These automated methods could potentially overcome the limitations of labor-intensive, qualitative, and descriptive approaches, offering improved efficiency and accuracy in the analysis of pathology images. In other words, it is impractical if the deep learning modeling process requires heavy efforts on data collection, data annotation, and computational resources to perform each specific learning task, considering the diverse image analysis tasks conducted in the digital pathology domain. Therefore, to further reduce the data collection effort and alleviate the above issues, we investigate unsupervised domain adaptation (UDA) and large-scale pre-trained vision-language models (VLMs), which have shown promising transferability across various downstream tasks.

## 1.2 Contributions

### 1.2.1 Supervised learning framework for pathology image analysis

In Chapter 2, an automated segmentation pipeline was proposed and evaluated on an Amyloid- $\beta$  stained WSI dataset [15, 26]. This pipeline includes a Convolutional Neural Network (CNN) module for segmenting GM/WM and a post-processing module used for removing artifacts and generating XML annotations that can be visualized and displayed on original-size WSIs via Aperio ImageScope. For the GM/WM segmentation module, we proposed ResNet-Patch that is memory-efficient and can achieve superior Intersection over Union (IoU) compared to U-Net [27] and FCN [11] in both GM and WM regions. Furthermore, we integrated a neural condition random field module that incorporates spatial correlations among neighboring patches into ResNet-Patch [26]. After that, we applied Gradient-weighted Class Activation Mapping (Grad-CAM) [28] to illustrate what CNN learns and relies on to distinguish GM from WM and confirm they are explainable from a pathologic perspective by consulting a domain expert.

On the other hand, Tang *et al.* proposed another CNN architecture and has shown its promise capability of identifying plaques associated with AD comparable to an expert [16]. However, it required the trained personnel to annotate more than 30,000 images for achieving such performance. We combined these two pipelines into one pipeline to detect and quantify the number of each plaque in GM and WM regions separately and visualize their density and distribution [25, 29].

### 1.2.2 Enhanced semi-supervised learning for practical scenarios

The proposed supervised learning pipelines help relieve the manual work after the deep learning model is well-trained. However, one obvious limitation is they still heavily rely on the number of labeled samples when training a CNN model, which unavoidably involves labor-intensive annotation costs. Meanwhile, their performances face severe degradation when labeled data is limited. To further minimize the manual efforts, we turned to semi-supervised learning (SSL), which leverages unlabeled data to improve the performance of supervised learning when the amount of labeled data is limited [30]. Consistency loss-based SSL methods involve pseudo labeling and data augmentation, showing their powerful performance on CIFAR-10 [17].

However, the applicability of these SSL algorithms in pathology images is still not fully explored. For example, a recent work [31] challenges the wide adoption of SSL in real-world scenarios, especially in class-imbalanced datasets [32, 33]. In Chapter 3 - Chapter 6, we propose four frameworks to enhance the robustness of SSL in various real-world settings.

#### **1.2.2.1 Improve pseudo-labeling for self-training**

One drawback of these consistency loss-based SSL methods is the imperfect class distribution is used to generate pseudo labels, and the over-reliance on pseudo labels makes it difficult to correctly update the class conditional distribution [34, 35]. For example, a recent study [21] applied FixMatch [17] and MixMatch [30] to a histology dataset and showed the performance of these state-of-the-art SSL methods are limited due to the above drawback.

To tackle this issue, inspired by [31], we proposed a novel two-stage SSL, SIM-FixMatch, to reduce the labeling cost when labeled data is too rare for transfer learning in Chapter 3.1 [36]. Our first stage is to employ self-supervised learning [37] for learning visual representations, which plays a similar role to pre-training in transfer learning but requires no labels. After the first stage, a pretrained encoder will be fed into the standard consistency loss-based SSL models. We first deployed FixMatch [17] to our WSI dataset as the baseline, and then evaluated our proposed framework on the same dataset: the results outperform FixMatch with the mean IoU score [38] at around 2% using 6,000 labeled tiles while over 10% by using only 600 labeled tiles from 2 WSIs.

#### **1.2.2.2 Class-imbalance semi-supervised learning**

To address the class-imbalanced issue, in Chapter 4, we first proved using smoothed weighting on the consistency loss is effective in countering the class imbalance problem. We then turned to the challenge of unknown distribution in unlabeled data. We proposed Smoothed Adaptive Weighting (SAW) to automatically estimate the class distribution of unlabeled data and assign weights to the consistency loss per epoch [39]. SAW forces the model to have a rough awareness of class distribution in unlabeled data and adjust the weights for all classes accordingly based on the model’s awareness. A more biased pseudo distribution will trigger a more vigorous response of weights derived from the model’s self-awareness. This self-adjusting scheme could gradually refine the pseudo labels to move closer to the true distribution of the unlabeled data.

We demonstrated the compatibility of SAW to any existing pseudo label-based SSL algorithms and the effectiveness of SAW under various imbalanced degrees among the labeled and unlabeled sets. Our algorithm improves both ReMixMatch [40] and FixMatch [17] by up to 30.0% and 40.5% of absolute geometry mean (GM) gain respectively. Furthermore, our framework outperforms DARP [32] (Distribution Aligning Refinery of Pseudo-label), a recent work that specifically tackles the class imbalance problem in SSL, by up to 10.9%. To demonstrate the robustness of our algorithm under different distributions, we also evaluated it on a hold-out test set with a completely different distribution from the training set and achieve up to 15% of absolute performance gain over DARP. To the best of our knowledge, our proposed method is the first work that eliminates the assumption of similar distributions between labeled and unlabeled data in class-imbalanced SSL [41, 42].

### 1.2.2.3 Joint semi-supervised and active learning with cost-effective labeling

Although we have proved the effectiveness of our proposed SSL framework in Chapter 3, one challenge is the selection criterion for labeled data for SSL in pathology images remains unclear. There are no well-established SSL guidelines on how to determine the amount of labeled data that is needed. Active learning (AL), on the other hand, aims at maximizing a model’s performance with minimal labeling efforts [43]. One challenge of AL is the *cold start problem*. When the selection of the starting set is insufficient, or the size is not sufficiently large, the models trained in subsequent cycles can be high-biased, resulting in poor selection again in the following cycles [44]. Another challenge is the annotation acquisition during the training process may require substantial rounds of labeling. This issue would notably extend the time needed for the entire project.

To minimize the labeling efforts, a natural direction is to combine AL and SSL, as both aim to improve a model’s performance on limited labeled data. More importantly, SSL and AL are complementary in function: SSL has the potential to leverage the unlabeled data to relieve the *cold start problem* of AL; on the other hand, AL can select the most optimal data to label for SSL. Therefore, in this work, as shown in Chapter 5, we combined SSL and AL to construct a label-efficient learning framework with a well-defined selection of regions for annotation to construct a small but effectively labeled dataset. To tackle AL’s “cold-start” issue, we designed a region-



based selection criterion to expand the diversity and volume of the labeled set effectively [45]. Finally, we evaluated the proposed framework on the same pathology segmentation task. With only 0.1% regions labeled, our proposed algorithm can reach a competitive IoU score compared to fully supervised learning and outperform the current state-of-the-art SSL by more than 10% of IoU score [38] and DICE coefficient [46]. This work is listed in Chapter 5.

#### **1.2.2.4 A Deployable Semi-supervised Learning with Informative Active Annotation for Pathology Image Classification and Segmentation**

Although SSL and AL have a similar goal of reducing labeling costs, only a few studies have attempted to combine them into a unified framework. In Chapter 6, we introduce Semi-Path, a practical SSL framework enhanced with active learning (AL) for gigapixel pathology tasks. Unlike existing methods that treat SSL and AL as independent components where AL incurs significant computational complexity to SSL, we propose a deep fusion of SSL and AL into a unified framework. Our framework introduces Informative Active Annotation (IAA) that employs an inner-outer two-loop structure to effectively extract knowledge from unlabeled pathology data. This structure significantly minimizes labeling efforts and computational complexity. Then, we propose Adaptive Pseudo Labeling (APL) to address heterogeneity in class distribution and prediction difficulty that are often observed in real-world pathology tasks. We evaluate Semi-Path on pathology image classification and segmentation tasks over three datasets that include WSIs from breast, colorectal, and brain tissues) and show the consistent superiority of Semi-Path over state-of-the-art methods.

### **1.2.3 Unsupervised domain adaptation**

Unsupervised domain adaptation (UDA) proposes to reduce data annotation costs by leveraging a labeled source domain to transfer the knowledge into an unlabeled target domain [47, 48, 49, 50, 51]. Prior UDA works focus on bridging the domain gap between source and target domains [51, 52, 53, 54], or increasing network capacity [49, 48] by changing a convolutional neural network (e.g., ResNet [55]) to Vision Transformer (ViT) [56]. All of these past methods are pre-trained on ImageNet [57], but large-scale pre-training is becoming practical and achieves superior performance in many fields [58, 59, 60, 61]. In theory, if the pre-trained dataset is large enough, the domain gap between source and target domains could be bridged by the pre-trained

dataset itself. Hence, we argue that large-scale pre-training is an important missing part of UDA in Chapter 7.

We attempt to fine-tune CLIP on the UDA dataset with a lower learning rate and observe less catastrophic forgetting issues, but the low learning rate prevents CLIP from learning new knowledge. To solve this problem, we propose to adjust the learning rate with Catastrophic Forgetting Measurement (CFM, Fig. 7.2) by comparing the original representation (forward original image on original CLIP) and fine-tuned representations (forward augmented images on fine-tuned CLIP). CFM is, however, unstable because every image has a different forgetting rate, so we leverage our observation that CLIP is likely to have similar predictions across all augmentations when the training example is easy (and large difference for the hard example). We propose to measure the consistency between weak (translate, flip) and strong augmentation (perturb visual appearance) as a momentum (Dual Consistency Momentum, DCM) to stabilize CFM. Our method does not introduce extra overhead: since the augmentation is already a part of fine-tuning, original prediction can be cached, and we do not need to fine-tune UDA and CLIP datasets jointly.

We further seek to use CLIP with pseudo-labeling on the target domain, which recently enjoyed success in UDA [62, 63, 49, 64, 63]. DebiasPL [65] utilized CLIP for pseudo-labeling, but it was designed for a single domain. After extending to source and target domains (UDA setting), DebiasPL[65] suffers from catastrophic forgetting issues (accuracy decreases by 21% on VisDA-2017 after fine-tuning). To solve this problem, we replace the fixed debias factor in DebiasPL[65] with our CFM, and replace the fixed momentum in DebiasPL with our adaptive momentum (DCM). We further include a domain name into a prompt (such as: “This is a **[sketch]** photo of [car]”). Our method mitigates the catastrophic forgetting issue, and achieves the best results on DomainNet [66], VisDA-2017 [67], Office-Home [68], Office-31 [69]. To summarize, our main contributions are:

- We propose to use CLIP in UDA and discover the catastrophic forgetting issue when fine-tuning CLIP. We propose CFM for CLIP in UDA to mitigate this issue without introducing extra computational overhead.
- We propose pseudo-labeling for CLIP in UDA by extending DebiasPL to multiple domains,

and replacing debias factor and momentum with our CFM and DCM. We also introduce a domain name into a prompt.

- We achieve the best results on four benchmarks on both ResNet and ViT, with a large performance improvement on the large-scale dataset (+18.5% accuracy on DomainNet) [70].

#### **1.2.4 Efficient adaptation of CLIP for data-efficient learning**

Deep learning with better network designs and large-scale well-curated datasets has achieved significant performance improvement in pathology image analysis tasks [29, 36]. However, collecting high-quality datasets with reliable annotations for each individual vision task may be time-consuming and labor-intensive [71, 72]. This may prevent the broad adoption of advanced deep learning techniques. To relieve the reliance on such datasets, pre-training and fine-tuning methods (TFT) have been studied in vision tasks: pre-train the model on a large-scale dataset and then fine-tune the model on different downstream tasks [73]. There are several challenges of such methods: 1) they may still require a large amount (over 10k) of labeled set to avoid the overfitting issue when fine-tuning the model for the downstream task [74, 75]; 2) the fine-tuning may not bring satisfactory performance in the target domain due to the existence of a large domain gap between the pre-trained data and pathology images [76]. Therefore, a natural question is how to efficiently fit the model to the new task with minimal overfitting when the new dataset is small.

In Chapter 8, we aim to fine-tune CLIP efficiently with light computing resources for pathology image analysis tasks. There are several challenges in fine-tuning CLIP. First, overfitting is a severe issue if we adapt CLIP to the downstream tasks directly since CLIP is pre-trained on a 400M dataset while the new domain dataset can be small [77]. Second, CLIP has shown its promising zero-shot generalization ability, directly fine-tuning it may lose it and face the “catastrophic forgetting” issue. In other words, it is unclear how to effectively learn the new knowledge while retaining the original pre-trained knowledge so that the generalization of CLIP can be maintained. Third, there are limited studies on benchmarking CLIP’s transferability in the pathology domain; hence, its applicability remains unclear. To address the above issues, we first study the applicability of CLIP on two public pathology datasets and benchmark the zero-shot ability of CLIP on them. Then we did empirical experiments to verify the importance of the CLIP’s pre-trained knowledge. To address the “catastrophic forgetting” issue, we propose

Residual Feature Connection (RFC) as a lightweight approach for adapting CLIP to pathology images. In RFC, we design a self-adaptive module to automatically fuse/balance the original knowledge from CLIP and the new knowledge learned from the new pathological task with only a few additional trainable weights instead of optimizing the entire encoders in CLIP. To further alleviate the overfitting, we also incorporate Hidden Representation Perturbation (HRP) and Dual-view Vision Contrastive (DVC). Lastly, we build an end-to-end adaptation of CLIP with a proposed loss, Doublet Multimodal Contrastive Loss (DMCL), to pathology tasks, named Path-CLIP [78].

We summarize our contributions as follows:

- We explore the applicability of CLIP on pathology images and highlight the catastrophic forgetting issues when adapting CLIP to downstream pathology tasks
- We propose Path-CLIP to adapt CLIP to pathology tasks with limited data. In Path-CLIP, we propose Residual Feature Connection (RFC) with Hidden Representation Perturbation (HRP) and a self-adaptive ratio to alleviate the overfitting and catastrophic forgetting issues. We also propose Dual-view Vision Contrastive (DVC) and Doublet Multimodal Contrastive Loss (DMCL) to fine-tune CLIP as an end-to-end pipeline.
- Extensive experiments demonstrate the efficiency and robustness of Path-CLIP. It has the potential to quickly adapt pre-trained CLIP to multi-tasks with competitive performance and light computational cost.

### 1.2.5 List of proposed technical methods

We list the abbreviations and their full names or explanations in Table 1.1.

### 1.2.6 List of publications

Here are the published first-authored publications.

1. **Z. Lai**, S. Vesdapunt, N. Zhou, J. Wu, X. Li, C. Huynh, C-N. Chuah, “PADCLIP: Pseudo-labeling with Adaptive Debiasing in CLIP for Unsupervised Domain Adaptation,” IEEE/CVF International Conference on Computer Vision (ICCV) 2023.

Table 1.1: List of proposed technical methods in this dissertation.

|           |                                                                                 |
|-----------|---------------------------------------------------------------------------------|
| BrainSec  | An automated pipeline for grey/white matter segmentation in brain tissue slides |
| CRF       | Conditional Random Field                                                        |
| SAW       | Smoothed Adaptive Weighting                                                     |
| IAA       | Informative Active Annotation                                                   |
| APL       | Adaptive Pseudo Labeling                                                        |
| CFM       | Catastrophic Forgetting Measurement                                             |
| DCM       | Dual Consistency Momentum                                                       |
| RFC       | Residual Feature Connection                                                     |
| HRP       | Hidden Representation Perturbation                                              |
| DVC       | Dual-view Vision COntrastive                                                    |
| DMCL      | Doublet Multimodal Contrastive Loss                                             |
| Semi-Path | A deployable pipeline for semi-supervised learning in pathology tasks           |
| CLIPath   | A pipeline to adapt CLIP for pathology images                                   |

2. **Z. Lai**, C. Wang, H. Gunawan, S-C. Cheung, and C-N. Chuah, “Smoothed Adaptive Weighting for Imbalanced Semi-Supervised Learning: Improve Reliability Against Unknown Distribution Data,” the 39th International Conference on Machine Learning (**ICML**) 2022. (**Won Participation Grant**)
3. **Z. Lai**, C. Wang, S-C. Cheung, and C-N. Chuah, “SaR: Self-adaptive Refinement on Pseudo Labels for Multiclass-Imbalanced Semi-supervised Learning,” 2022 **CVPR** Workshop on Learning with Limited Labelled Data for Image and Video Understanding. (**Best Paper Award**)
4. **Z. Lai**, L. Cerny Oliveira, R. Guo, W. Xu, Z. Hu, K. Mifflin, C. DeCarlie, S-C. Cheung, C-N. Chuah, and B. N. Dugger, “BrainSec: Automated Brain Tissue Segmentation Pipeline for Scalable Neuropathological Analysis,” **IEEE Access**, 2022.
5. **Z. Lai**, Z. Li, L. Cerny Oliveira, J. Chauhan, B. Dugger, and C-N. Chuah, “CLIPath: Fine-tune CLIP with Visual Feature Fusion for Pathology Image Analysis Towards Minimizing Data Collection Efforts,” 2023 **ICCV** 2nd Workshop on Computer Vision for Automated

Medical Diagnosis (CVAMD).

6. **Z. Lai**, C. Wang, L. Cerny Oliveira, B. Dugger, S-C. Cheung, C-N. Chuah, “Joint Semi-supervised and Active Learning for Segmentation of Gigapixel Pathology Images with Cost-Effective Labeling,” 2021 **ICCV** Workshop on Computational Challenges in Digital Pathology (CDpath).
7. **Z. Lai**, C. Wang, Z. Hu, B. Dugger, S-C. Cheung, C-N. Chuah, “A Semi-supervised Learning for Segmentation of Gigapixel Histopathology Images from Brain Tissues,” 43rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (**EMBC**), 2021.
8. **Z. Lai**, P. Vadlaputi, D. J. Tancredi, M. Garg, R. I. Koppel, M. Goodman, W. Hogan, N. Cresalia, S. Juergensen, E. Manalo, S. Lashminrusimha, C-N. Chuah, and H. Siefkes, “Enhanced Critical Congenital Cardiac Disease Screening by Combining Interpretable Machine Learning Algorithms,” 43rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (**EMBC**), 2021.

# Chapter 2

## BrainSec: Supervised Learning Pipeline for Pathology Image Analysis

As neurodegenerative disease pathological hallmarks have been reported in both grey matter (GM) and white matter (WM) with different density distributions, automating the segmentation process of GM/WM would be extremely advantageous for aiding in neuropathologic deep phenotyping. Standard segmentation methods typically involve manual annotations, where a trained researcher traces the delineation of GM/WM in ultra-high-resolution Whole Slide Images (WSIs). This method can be time-consuming and subjective, preventing a scalable analysis on pathology images. This paper proposes an automated segmentation pipeline (BrainSec) combining a Convolutional Neural Network (CNN) module for segmenting GM/WM regions and a post-processing module to remove artifacts/residues of tissues. The final output generates XML annotations that can be visualized via Aperio ImageScope. First, we investigate two baseline models for medical image segmentation: FCN, and U-Net. Then we propose a patch-based approach, BrainSec, to classify the GM/WM/background regions. We demonstrate BrainSec is robust and has reliable performance by testing it on over 180 WSIs that incorporate numerous unique cases as well as distinct neuroanatomic brain regions. We also apply gradient-weighted class activation mapping (Grad-CAM) to interpret the segmentation masks and provide relevant explanations and insights. In addition, we have integrated BrainSec with an existing Amyloid- $\beta$  pathology classification model into a unified framework (without incurring significant

computation complexity) to identify pathologies, visualize their distributions, and quantify each type of pathologies in segmented GM/WM regions, respectively.

In this chapter, we propose an entire pipeline used for segmenting GM/WM regions with a post-processing module that removes artifacts and unwanted fragments. Finally, the pipeline generates XML annotations that can be displayed on the original WSIs. This pipeline is also able to incorporate exterior CNN modules that perform other tasks in the same region. We implement two baseline CNN models that are widely deployed in medical image segmentation problems, namely a fully convolutional network (FCN) [79] and a U-Net model [80], to perform pixel-wise segmentation of GM and WM. Then, we propose BrainSec, a novel mechanism that transforms the pixel-wise segmentation problem of gigapixel WSIs into a patch-wise classification problem in the first stage and then subsequently converts it back to the original segmentation task according to various resolution requirements of outputs. To further enhance the proposed workflow, we incorporated BrainSec into an existing pipeline that quantified amyloid pathologies in select brain areas [16]. To alleviate the prediction inconsistency issue in the first stage of BrainSec, we incorporate BrainSec with a Neural Conditional Random Field (NCRF) module to model the spatial relationships among neighboring patches, referred to as BrainSec+. BrainSec+ merges two modules into one by achieving a similar effect of the post-processing module.

Our contributions can be summarized as follows:

- We establish an automated GM/WM segmentation pipeline, BrainSec. To omit the need for an additional post-processing step, we propose BrainSec+ based on BrainSec.
- We apply Gradient-weighted Class Activation Mapping (Grad-CAM) [28] to confirm what the model learns is explainable from a pathologic perspective by consulting a domain expert in neuropathology.
- We demonstrate that BrainSec is consistently performing well over more than 180 WSIs (visually inspected by trained personnel) for a broader neuroanatomical evaluation.
- We show the compatibility of BrainSec into another neuropathology detection model by incorporating BrainSec into an existing pipeline to aid in the quantification of neuroanatomic specific data on Amyloid- $\beta$  plaques.



- After the integration, we propose an entire framework to quantify the number of neuropathological hallmarks and visualize their distribution in GM/WM regions.

## 2.1 Related work

### 2.1.1 Image processing-based segmentation

Although many methods [81, 82] have shown their success in Magnetic Resonance Images (MRIs), they may not be applicable to gigapixel-pixel WSIs considering the distinct imaging processes and visual differences between MRIs and WSIs [83]. For WSIs, Hiary *et al.* [84] built an automatic algorithm based on k-means clustering that worked on pixel intensity and texture features. Bug *et al.* [12] applied global thresholding at the mean value of the Gaussian blurred Laplacian of the greyscale image. In [85], stain concentration and porosity analysis were applied to the segmentation of H&E stained WSIs. Although these methods are computationally efficient, the segmentation results are very limited due to inter and intra-variations in staining and color contrast, resulting in general failure for the above-mentioned methods on hold-out test sets [27]. Additionally, these methods have only been tested on non-brain WSIs like breast, tongue, and skin, so their applicability and feasibility on our brain WSIs have yet to be determined. In this work, we first investigated the above methods on our slides, but they may not generalize well on hold-out test sets and need to be fine-tuned for heterogeneous settings (e.g., different scanners, different brain regions, etc.). Hence, we turn our attention to deep learning-based methods to seek a more generalizable approach.

### 2.1.2 Deep learning-based segmentation

In recent years, deep learning methods have gained vast popularity in image segmentation problems in pathology involving use of WSIs. In [86], CNNs and Image Processing methods are compared for segmenting invasive ductal carcinoma in breast tissue slides. Breast tissue WSIs were also used by [87] to achieve segmentation of tumor tissue through use of DCNNs. In [88], a ResNet model pretrained on ImageNet was tuned through a Transfer Learning approach to segment the mitotic cells in pre-processed breast tissue WSIs. Recently, in [89], a pipeline powered by VGG16 was used to segment cancer tissue in bladder WSIs and grade the severity of cancerous tissue.

Besides these CNN methods, FCN [11] and U-Net [27] based architectures are predominant choices in medical image segmentation problems [80, 90]. The study from [91] is an example of U-Net being applied for nuclei segmentation from glioma patients' WSIs. In [92], we see a modified version of U-Net, Y-Net, achieve state-of-the-art segmentation of whole slide breast cancer biopsy images. A multi-scale U-Net was also used by [93] to segment whole slide prostate images in order to identify and grade cancerous tissue. Some studies have used U-Net in medical image segmentation involving the brain, but using data different than WSI [94, 95, 96]. So far, only one study has investigated these architectures on GM/WM segmentation in gigapixel pathology images [97]. However, the study used 49 Tau-stained WSIs, and only FCN architecture was used. Here, we examined multiple architectures and compared their performance. The details of how we implement them as benchmarks are as follows.

**FCN.** Fully convolutional network (FCN), due to its arbitrary input size and better localization performance [98], was a popular choice in the recent works of medical image segmentation [99]. When applied to tissue segmentation of histopathological WSIs, FCN outperforms traditional methods and achieves fewer outliers and more stable results [11]. FCN is used for multi-organ segmentation[100], in both 2D slices of 3D volume [101] and 3D images [102]. Besides regular FCNs, modified Cascaded FCNs have been deployed in liver segmentation, where the first FCN performed Region of Interest (ROI) and the second performed the lesion segmentation [103]. Multi-stream FCN can improve accuracy by allowing the system to take multiple forms of the same organ [104]. Moreover, focal loss is applied to the FCN due to the class imbalances of training data. The number of false positives can be removed by applying the focal loss on the FCN network [105, 106].

The most recent FCN model (as shown in Figure 2.1) we explored is based on the architecture and hyper-parameters in [11]. We used FCN-8s model from the original FCN paper [98], which combined predictions from the final layer, pool4 layer and pool3 layer at the stride of 8. The FCN architecture contains 7 convolutional layers. The first two layers have the filter size  $5 \times 5$ . The third and fourth layers have filter size  $3 \times 3$ . The fifth layer has the size of  $11 \times 11$ . The last two layers have  $1 \times 1$  convolution, which are equivalent to the fully connected layers. The number of filters at each layer are 16, 32, 64, 64, 1024, 512 and 2 respectively as shown in Figure 2.1. Filter

size of  $2 \times 2$  and stride of 2 are used to all the max pooling layers. The number of max pooling layers is based on original FCN paper [98], which are inserted after first five convolutional layers to reduce memory requirements of the network. The batch normalization layer and sigmoid activation layer are added after every convolutional layer for the regularization and convergence purposes. Drop out layer with drop out rate of 0.2 is added after every convolutional layer to reduce the probability of overfitting. The batch size is 128.

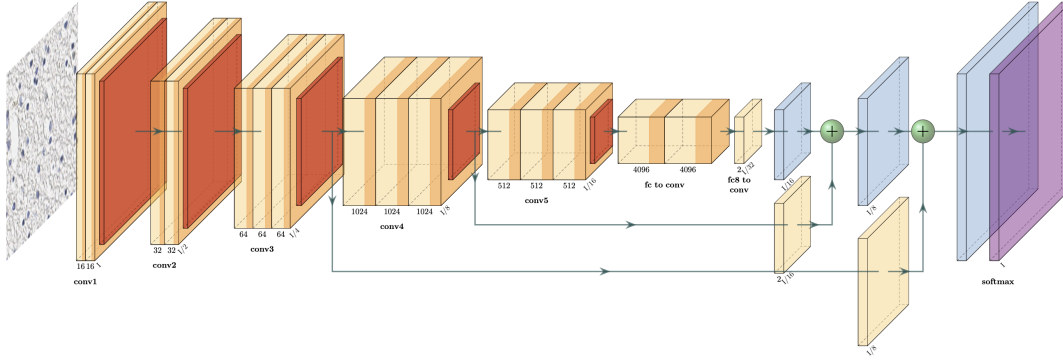


Figure 2.1. FCN Architecture.

By benchmarking the results of different loss functions, we chose the combination of the categorical cross entropy loss and focal loss for FCN [105, 106]. By setting the weight of different output classes to be one and training FCN on categorical cross-entropy loss and then on focal loss, the average validation results of three output classes outperformed using either categorical cross-entropy or focal loss alone (weighted or unweighted).

**U-Net.** U-Net is one of the state-of-the-art models in medical image segmentation and has gained great popularity in different medical problems. For example, Dong et al. [107] built a fully automatic method based on U-Net for brain tumor segmentation of MRI. In [108], Rampun et al. proposed a 2D fetal brain segmentation of MRI by modifying the U-Net architecture. A major drawback of U-Net based methods is that they require computationally intensive operations on GPUs and take a great deal of processing power with very trivial gain in performance [109].

The updated U-Net model we investigated is based on the architecture and hyper-parameters in [27]. As shown in Figure 2.2, the contracting path of the network has four convolution levels. Each level consists of two consecutive  $5 \times 5$  convolution operations (zero-padded convolutions)

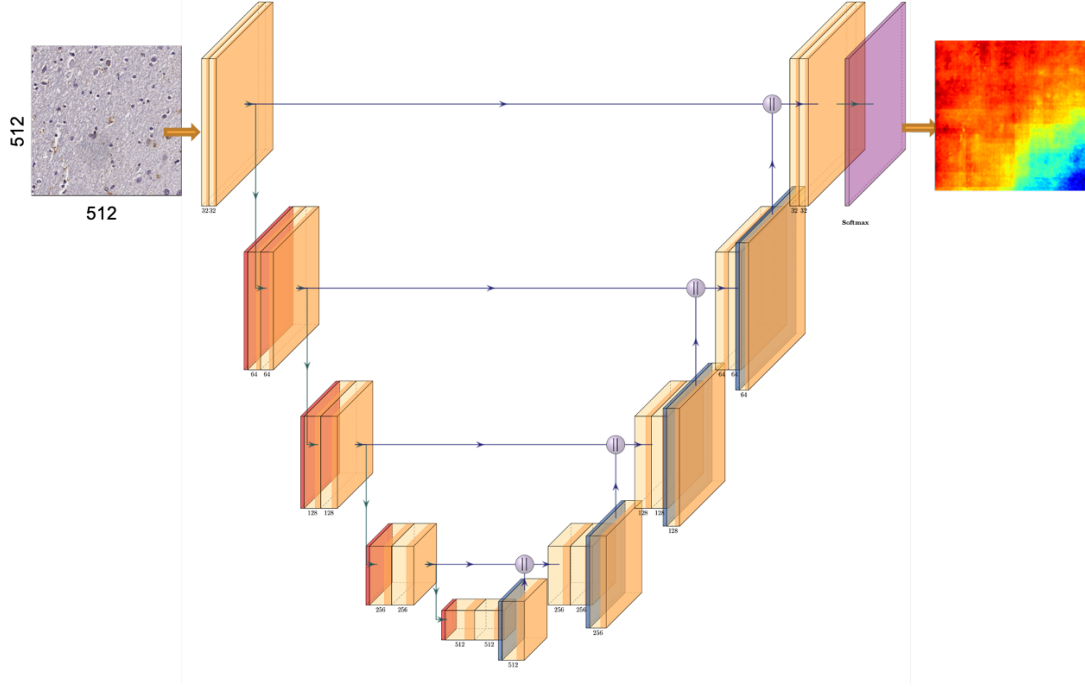


Figure 2.2. U-Net Architecture.

with Exponential Linear Unit (ELU) [110] activation functions, followed by a  $2 \times 2$  max-pooling operation with stride 2 for down-sampling. At each down-sampling level, the number of feature channels is doubled ( $32 - 64 - 128 - 256 - 512$ ). The expansive part of the network also has four convolution levels. Each level consists of a  $2 \times 2$  transposed convolution that halves the number of feature channels, a concatenation with the corresponding contracting level output, and two  $5 \times 5$  convolutions with ELU activation. To speed up learning and provide some regularization effect, we used batch normalization [111] after each convolutional layer. We also incorporated drop-out [112] after the first convolutional layer at each level, both in the contracting and the expansive path. A final  $1 \times 1$  convolution with 3 output channels is then used to map the last feature map to the class prediction output.

Prior to the training, all convolution kernels were initialized using *He's uniform* initialization [113]. We used the *Adam* optimizer [114] with an initial learning rate of 0.001. The  $\alpha$  parameter for the ELU activations was set to 1.0, the drop-out rate was 0.2, and the batch size was 16. To diminish the effect of class-imbalance issue in our dataset, we weighted the categorical cross entropy loss function by the inverse of class frequencies in the training dataset.

However, there is another obvious limitation among these deep learning-based methods: these networks were specifically designed and tested on specific medical applications such as tumor detection or image modality (MRI that has standard image formats and CT), making it difficult to generalize existing findings to our human brain tissue WSIs that have no standard formats [115]. While MRI and CT are representations of a global view of the whole brain, our WSIs can only capture a local view of one region of brain, unlike semantic objects in natural images, there are subtle and gradual changes of pathological features rather than the sharp color edges in clinically meaningful regions. Many CNN models rely heavily on annotations, indicating the difficulty for them to self-identify residues and artifacts in the tissue slides if the annotations are not provided. For example, in [15], although the CNN architecture is able to distinguish GM from WM, it is unable to remove residues and artifacts since the corresponding annotations were not provided.

### **2.1.3 Patch-based approach for ultra-high resolution images**

Most of these deep learning methods have only been tested on images with low to medium resolution of up to a few megapixels. Chen et al. [14] regarded images with resolution up to 30 million pixels as ultra-high resolution images. They proposed a method that integrates both global downsampled images and local patches while they only tested their methods in images with resolution up to 30 million pixels. However, the average resolution of our WSIs exceeds 2 gigapixels. Downsampling used in [14] cannot be directly transferred to our case as we have only 30 WSIs with expert annotations, which indicates the shortage of training data, while the datasets used in their experiments consist of more than 2,000 images. Besides, we need to preserve minute details of pathologies (such as plaques) and cells in our WSIs while downsampling will unavoidably lose these important features. In [15], the authors conducted their experiments by introducing downsampling in the training process, which resulted in their limited performance on final masks.

Because of the ultra-high resolution and details that needs to be preserved, recently many studies applied patch-based methods to WSIs [116, 117, 118]. They extracted small patches (e.g.  $256 \times 256$  pixels) from WSIs and then trained a deep CNN model. Although these studies solved the resolution issue, most of these applications are classification problems (e.g. breast cancer

classification [118]) while our problem is a segmentation problem. Besides, as the patches were extracted and trained independently in these studies, the spatial correlations shared by small patches and their neighbors may not be modeled explicitly. As a result, the predictions over neighboring patches may not be consistent and there may exist isolated outliers in the patch-level probability map during inference time [119, 120]. To incorporate the spatial correlations between neighboring patches, authors in [119] used Conditional Random Field (CRF) to model spatial correlations and refine the predicted probability map in the post-processing stage. However, due to the high computational overhead, the authors can only feed five features from patch representations into the CRF inference algorithm. In this paper, we propose a pipeline that first converts the GM/WM segmentation to a patched-based classification problem, and then generates pixel-level segmentation output based on the classification results. Subsequently, we propose a one-stage segmentation method that incorporates spatial correlations among neighboring patches through a Neural Conditional Random Field (NCRF) [121] layer.

## 2.2 Datasets

### 2.2.1 30-WSI Dataset

To provide quantitative results and initially test, train, and validate our pipeline, we first evaluated a 30 Whole Slide Images (WSIs) dataset annotated using QuPath[5]. These WSIs were Amyloid- $\beta$  stained [122] and were from formalin-fixed paraffin-embedded 5um sections of the temporal cortex. Aperio AT2 digitized these WSIs at up to  $40\times$  magnification. The resolution is nearly  $60,000 \times 50,000$  pixels each on average. These 30 WSIs can be split into two sets: one set includes 18 cases that had a pathologic diagnosis of AD while the other set (12 cases) lacked a pathologic diagnosis of AD but may have other diseases, such as vascular disease (Non-Alzheimer disease, NAD). The ethnoracial makeup of the cohort was 3 Hispanics (10%) , 5 African Americans (17%), and 22 non-Hispanic White (73%) descendants.

These studies utilized tissues only from human post-mortem. Only living subjects are confirmed as Human Subjects under federal law (45 CFR 46, Protection of Human Subjects). All participants or legal representative approved informed consent during the life of the participant as part of the University of California Davis Alzheimer’s disease Center program. The data

collection process followed current laws, regulations and IRB guidelines. All of these 30 WSIs had been de-identified, lacking personal health information like names, social security numbers, addresses, and phone numbers. To further protect data confidentiality, we named the AD cases as WSI-1 to WSI-18 and NAD cases as WSI-19 to WSI-30.

### 2.2.2 Training data preparation

To deal with the issue of ultra-high resolution, we use Pyvips Library [123] to set up image processing pipelines on original-resolution WSIs instead of directly manipulating these slides. Hence, we avoided loading the entire image into memory at once for processing because Pyvips can stream the image in parallel from the first step to the last step of pipelines simultaneously. The proposed approach in [15] manually selected regions from GM, WM, and background and cropped small tiles from these regions separately. This manual action could result in limited variety of datasets and involve human interventions. As we have pixel-wise annotations on GM and WM, we randomly cropped patches  $256 \times 256$  from the slides in the training set (20 slides) and labeled them as follows: each patch was labeled with the category of the central pixel of that patch tracing back to our pixel-wise ground truth. After this, we obtained around 310k patches from GM, 100k patches from WM, and 300k patches from background. In our experiments, 20 WSIs (12 AD cases and 8 NAD cases) were randomly selected for training and validation while the remaining 10 WSIs (6 AD cases and 4 NAD cases) were used for **hold-out** testing and inference. The data split was fully random as GM region and WM region are in the same level of magnitude (the ratio between GM/WM region in one slide ranges from 3:1 to 1:1).

For the annotation process, we first invited one trained personnel (K.M) to draw the boundary between GM and WM. Then we sent these annotations to an expert (B.D) for checking and making the final annotation. We use the annotation verified by the expert as ground truth.

### 2.2.3 130-WSI and 52-Heteroarea Datasets

In order to rigorously test the performance of our segmentation model, two additional datasets were prepared from available convenience samples. The 130-WSI dataset contained temporal WSIs from 130 new cases not included in the 30-WSI dataset. This dataset aimed at testing our model on as many new WSIs as possible to check if performance is affected by applying

it to an unseen, larger cohort. There were 13 NAD cases and 117 AD cases in this set. The ethnoracial make up of the cohort was 13 Hispanics (10%) , 15 African Americans (11.5%), 6 Asian Americans (4.6%), 1 Other (0.8%) and 95 non-Hispanic White (73.1%) descendents.

The other additional dataset, 52-Heteroarea, contained 52 new WSIs from two brain areas not seen by the model: areas from frontal and parietal lobes. The goal of this dataset was to test our model performance on brain areas different than the temporal region. The 52-Heteroarea contained 26 cases, with 13 cases featured in the 130-WSI dataset, 2 cases featured in the 30-WSI dataset and 11 new cases. From each of the 26 cases, 2 WSIs were acquired, one from each unseen brain area (frontal and parietal cortex sections). The summary of the information on each of the three datasets used can be seen on Table 2.1. For both 130-WSI and 52-Heteroarea datasets the procedure for staining, scanning, and preprocessing was the same as the 30-WSI dataset. All protocol regarding de-identification and confidentiality followed by the 30-WSI dataset was also followed for the additional two datasets. These additional datasets were treated as hold-out sets, and no ground truth GM/WM annotations were collected. The goal was to perform a proof of concept by visually inspecting the BrainSec prediction results (background vs. GM vs. WM).

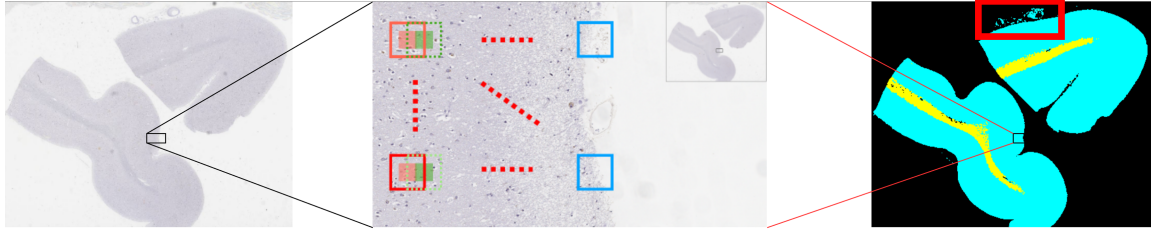
Table 2.1: Summary of the three datasets used in the study

| <b>Dataset</b> | <b>Number of cases</b> | <b>% of NAD cases</b> | <b>Brain Areas</b> |
|----------------|------------------------|-----------------------|--------------------|
| 30-WSI         | 30                     | 40.00%                | Temporal           |
| 130-WSI        | 130                    | 10.00%                | Temporal           |
| 52-Heteroarea  | 26                     | 15.38%                | Frontal, Parietal  |

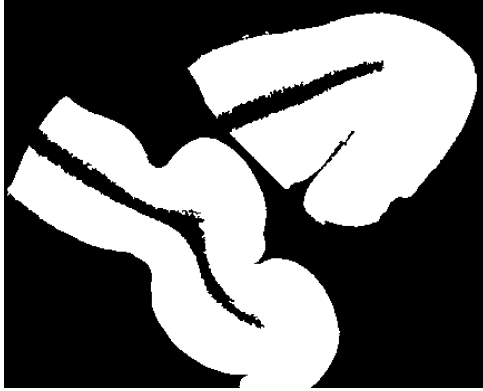
## 2.3 Methodology

In this section, we first introduce BrainSec, a GM/WM segmentation pipeline. Then, we present its integration into a patch-based pathology classification model [16] and propose a framework to automate the quantification of pathologies in GM/WM regions in a scalable way and visualize the distributions of each type of pathology, as shown in Figure 2.3. We release our codes on <https://github.com/ucdrubinet/BrainSec>.





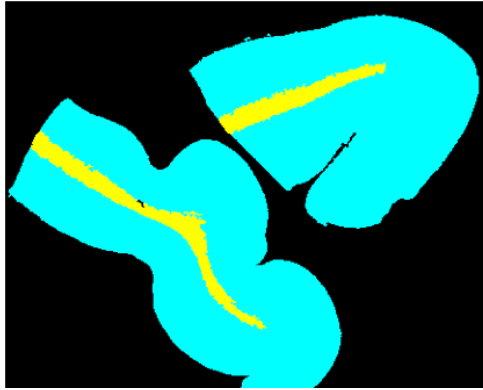
(a) Sliding Window to Generate Masks before Post-Processing



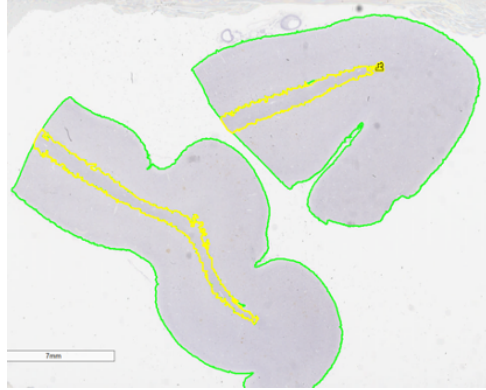
(b) GM Mask



(c) WM Mask



(d) After Post-Processing



(e) Final Output with XML annotations

Figure 2.3. Segmentation pipeline and the output of XML annotations.

### 2.3.1 BrainSec

BrainSec has two modules: CNN module and post-processing module. CNN module consists of two stages: transform the problem to patch-based classification and convert back to pixel-wise segmentation of WSIs.

### 2.3.1.1 Patch-based classification

In this stage, we transform the pixel-level segmentation problem into a patch-based classification problem. For image classification tasks, a myriad of different CNN architectures have been proposed, such as AlexNet, VGG, ResNet [124] in the past few years. While design motivations of each architecture are greatly different, there is increasing evidence showing that the features extracted from these state-of-the-art architectures are quite akin and their improvements gradually become trivial and start to converge [125]. To produce reasonable results with relatively minimized complexity, we decide to select a simple architecture to pursue the tradeoff between complexity and performance. In [55], He et al. showed their ResNet architecture has fewer filters and lower complexity but achieves similar performance compared to other state-of-the-art architectures. As such, we selected the ResNet-18 architecture described in [55] as our backbone network and the fundamental basis of our methods. We selected  $1 \times 1$  convolution skip connection instead of identity skip connection in our backbone.

We modified ResNet-18 by redefining the last fully connected layer to output three categories: GM, WM, and the background of tissue slides. As we transform the segmentation problem into a classification problem in this stage, the function of ResNet-18 here is to extract the features from each patch and classify its corresponding category. We adapted pre-trained parameters except for the last layer from ResNet-18 trained on ImageNet because it has already shown the ability of extracting useful features from natural images. We introduced categorical weights to the multi-class cross-entropy loss based on the inverse of class frequencies in our training set. Specifically, as we have around 310k patches from GM, 100k patches from WM, and 300k patches from background in our training set, the weight ratio of GM:WM:background is 1:3:1 (inverse of class frequencies). We used *Adam* optimizer [114] and the initial learning rate is 0.001. We adopted the automatic early stopping criteria by saving checkpoints for each epoch for both training and validation sets, subsequently selecting the well-trained model automatically. The batch size was set as 16.

### 2.3.1.2 Pixel-wise segmentation

In this stage, we utilize the results from the classification task to construct pixel-wise segmentation output. In the previous stage, each patch is classified to a corresponding output class

(GM, WM, or background) using our modified ResNet-18 model. In order to achieve pixel-wise segmentation results, we use a sliding window approach to extract each patch until the whole image is fully covered. The resolution of output masks will be decided by the step size. For example, as shown Figure 2.3a, step size here is set as 128 while the patch size is 256. The red patch in Figure 2.3a is fed into our modified ResNet-18 first and subsequently receives the prediction category. After that, if output masks are required to be the same size as original WSIs, all pixels in the central area (red block) with the size of  $\text{step\_size} \times \text{step\_size}$  ( $128 \times 128$  here) will be classified as the same category predicted by the modified ResNet-18 model. Then, we step forward with the step size to the green patch with a dotted line and repeat the same action so the green block next to the red block will be labeled with the same category. By repeating this action, after the sliding window traverses the whole image, we can get segmentation masks for GM, WM (as shown in Figure 2.3b, c), and background separately. As such, we can see the accuracy of output masks is determined by step size: if the step size is larger, outputs will be less accurate around boundaries, but inference complexity will be reduced; if the step size is smaller, outputs will be more accurate, but inference complexity is also increasing. Hence, BrainSec provides flexibility to achieve different trade-offs between performance and complexity. On the other hand, if the goal is to construct a framework for pathology quantification and visualization, the step size should be correlated with the size of a pathology. We expect the size of a sliding window to be adequately larger than the size of an Amyloid- $\beta$  deposit, thus selecting it as 128 to identify and precisely locate each pathology. The step size can be altered for other types of pathologies.

### 2.3.1.3 Post-processing

We design a post-processing module to refine our segmentation output masks and remove tissue residues in our WSIs. The full-resolution predicted outputs are down-sampled by a factor of 16 (1/4 in width and 1/4 in height) to reduce computational complexity. Two consecutive area openings are applied to remove fuzzy predictions of GM and WM with pixel area  $< 20,000$ , and an area closing is applied to remove fuzzy prediction of background with pixel area  $< 12,500$ . Then, small tissue residues with areas smaller than 5% of the WSI area are removed. In the end, a morphological opening with a disk-shaped kernel with a radius of eight was applied to

smooth the boundary, and the output was up-sampled back to the original resolution. The kernel sizes used for morphological opening and the size threshold for pixel areas removed are obtained empirically. Figure 2.3a, 2.3d is an example showing the difference between before and after the post-processing method is applied to remove tissue fragments.

In addition to these steps, we generate XML annotation files of GM and WM segmentation boundaries by finding contours of GM and WM segmentation masks, respectively as shown in Figure 2.3e. These XML files are in the same format as our ground truth annotations and can be visualized and displayed on original-size WSIs using Aperio ImageScope. They are also downsampled by  $50\times$  to smooth out the boundary for faster visualization.

### 2.3.2 BrainSec+

The post-processing module in Section 2.3.1.3 is effective for refining the segmentation output of BrainSec. However, it still requires parameter tuning, which may prevent it from the wide adoption of various pathology datasets. In this section, we propose BrainSec+, which is built upon BrainSec by using a similar mechanism and backbone as BrainSec while integrating a Neural Conditional Random Field (NCRF) layer that incorporates spatial relationships among neighboring patches to alleviate the need for a post-processing module. Conditional Random Field (CRF) BrainSec+ (as shown in Figure 2.4) consists of two components: ResNet-18 component is used to extract features and encode each patch as an embedding (a vector representation with a fixed length) by taking a grid of patches as input; NCRF layer is used to model spatial correlations among the grid of patches and provide an output of the probability of the central patch in the grid.

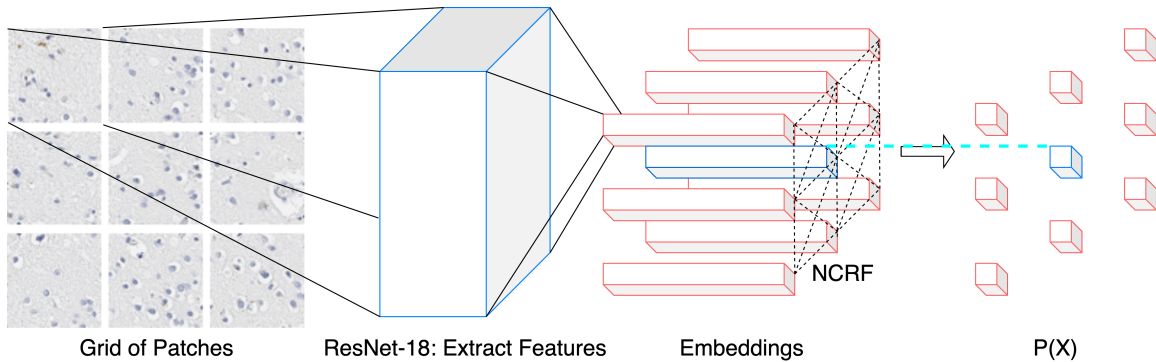


Figure 2.4. NCRF Architecture.

### 2.3.2.1 Spatial correlations with NCRF

Given a grid of patches, we define a random field  $X = \{X_1, X_2, \dots, X_N\}$  as the random variables related to each patch, where  $N$  is the number of patches in the grid, e.g. 9 for a grid of  $3 \times 3$ . Each random variable represents a label from the set of  $\{GM, WM, background\}$ , conditioned on observations  $I = \{I_1, I_2, \dots, I_N\}$ , where  $I$  is the embedding of each original patch extracted by the CNN component. This set of label is defined as  $x = \{x_1, x_2, \dots, x_N\}$ . Therefore, different from the CRF methods for pixel labeling,  $I$  is not a set of pixel attributes such as RGB color values or intensity but a set of patch descriptors: CNN features from each patch. The distribution of  $(X, I)$  can be a CRF if the random variables of set  $X$  conditioned on observations  $I$  meet the requirements of Markov property. Hence, the structured prediction can be transformed as:

$$y^* = \underset{x \in X}{\operatorname{argmax}} P(X|I) \quad (2.1)$$

where  $P(X|I)$  is defined as a Gibbs distribution:

$$P(X|I) = \frac{1}{Z(I)} \exp(-E(X|I)) \quad (2.2)$$

In a fully-connected CRF [126], the energy of Gibbs can be written as:

$$E(X|I) = \sum_i \psi_u(x_i|I_i) + \sum_{i < j} \psi_p(x_i, x_j|I_i, I_j) \quad (2.3)$$

where  $i, j \in \{1, 2, \dots, N\}$ . The energy function  $E(X|I)$  is the summation of two terms: a unary potential  $\psi_u(x_i|I_i)$  and the pairwise potential  $\psi_p(x_i, x_j|I_i, I_j)$ . The unary potential can be considered an initial estimate of the cost of patch  $i$  with its label  $x_i$  given its corresponding embedding  $I_i$ . Similarly, the pairwise potential estimates the joint cost of patch  $i, j$  with their label  $x_i, x_j$  given their corresponding embeddings  $I_i, I_j$ . In other words, the pairwise potential is a measure of relationships between two patches and is defined over every pair of patches in the grid. In [127], they use the cosine similarity to define the pairwise potential as:

$$\psi_p(x_i, x_j|I_i, I_j) = \mathbf{H}(x_i = x_j) \cdot w_{i,j} \left( 1 - \frac{I_i \cdot I_j}{\|I_i\| \cdot \|I_j\|} \right) \quad (2.4)$$

where  $\mathbf{H}(I_i = I_j)$  is an indicator function that checks the label compatibility between  $X_i, X_j$ . By using the cosine similarity, the spatial correlations between a pair of patches can be modelled by

encouraging lower cost in the case that  $X_i, X_j$  are assigned with the same label if the embeddings  $I_i, I_j$  are similar.  $w_{i,j}$  is a single, trainable parameter that the authors claimed related to spatial distance between patches [127].

### 2.3.2.2 End-to-End back-propagation

To compute the cross-entropy loss with the ground truth labels, we need to get the marginal distribution of each patch label  $x_i$  so that the standard back-propagation algorithms can be applied here to achieve end-to-end training. A mean-field approximation method was proposed in [128] to approximate maximum posterior marginal inference. As exact marginal inference is intractable, by using the mean-field approximation, the original CRF distribution  $P(X)$  can be transformed to  $Q(X)$ , which is obtained by the product of independent marginal distributions, as shown in (2.5):

$$P(X) \approx Q(X) = \prod_i^N Q_i(x_i) \quad (2.5)$$

Here  $Q(X)$  is a simpler distribution compared to  $P(X)$ . The  $\mathbb{KL}$  divergence between  $Q(X)$  and  $P(X)$  is shown in (2.6).

$$\mathbb{KL}(Q(X)||P(X)) = \sum_i Q(x_i) \log \frac{Q(x_i)}{P(x_i)} \quad (2.6)$$

The first item is the negative of entropy, while the second item is cross-entropy. Assuming  $\tilde{P}(X|I) = \exp(-E(X|I))$  as the unnormalized CRF distribution, we can derive the cross-entropy item as shown in (2.7).

$$\begin{aligned} \sum_i Q(x_i) \log P(x_i) &= \sum_i Q(x_i) \log \frac{\tilde{P}(x_i)}{Z} \\ &= \sum_i Q(x_i) \log \tilde{P}(x_i) - \sum_i Q(x_i) \log Z \\ &= \sum_i Q(x_i) \log \tilde{P}(x_i) - \log Z \end{aligned} \quad (2.7)$$

For minimizing  $\mathbb{KL}$  divergence,  $-\log Z$  can be dropped as this is an additive constant. This leaves us with the following optimization problem:

$$\begin{aligned} &\arg \min \mathbb{KL}(Q(X)||P(X)) \\ &= \arg \min [\sum_i Q_i(x_i) \log Q_i(x_i) - \sum_i Q_i(x_i) \log \tilde{P}(x_i)] \\ &= \arg \min \sum_i Q_i(x_i) (\log Q_i(x_i) - \log \tilde{P}(x_i)) \end{aligned} \quad (2.8)$$

As  $\mathbb{KL}$  divergence cannot be negative, the minimum of Equation 2.8 can be derived as:

$$\log Q_i(x_i) = \mathbb{E}_{-x_i}[\log \tilde{P}(X)] + \text{const} \quad (2.9)$$

This is the update equation for each marginal distribution  $Q_i(x_i)$ , where  $\mathbb{E}_{-x_i}[\log \tilde{P}(X)]$  refers to the expectation of  $\log \tilde{P}(X)$  on all  $x$  except  $x_i$ . In summary, our mean-field marginal inference algorithm is shown in Algorithm 1.

---

**Algorithm 1** Mean-field Marginal Inference Algorithm

---

```

for each  $i \in \{1, 2, \dots, N\}$  do
    calculate  $\psi_u(x_i|I_i)$ 
    initialize  $Q_i(x_i) \leftarrow \psi_u(x_i|I_i)$ 
    normalize  $Q_i(x_i)$ 
    for each  $j \in \{1, 2, \dots, N\}$  do
        calculate  $\psi_p(x_i, x_j|I_i, I_j)$ 
         $\log \tilde{P}(X|I) \leftarrow \sum_i \psi_u(x_i|I_i) + \sum_{i < j} \psi_p(x_i, x_j|I_i, I_j)$ 
    end for
end for

for T iterations do
    for each  $i \in \{1, 2, \dots, N\}$  do
         $\log Q_i(x_i) \leftarrow \mathbb{E}_{-x_i}[\log \tilde{P}(X|I)]$  normalize  $Q_i(x_i)$ 
    end for
end for

```

---

### 2.3.3 Pathology quantification & visualization with BrainSec

As BrainSec is more computationally efficient, we use this as one example to illustrate its compatibility with other patch-based pathology models. Based on BrainSec and the pathology model [16], we propose an entire framework (as shown in Figure 2.5) to quantify each type of Amyloid- $\beta$  deposit and visualize distributions in GM/WM regions. We denote BrainSec in this case as  $g(\cdot)$ .  $f(\cdot)$  [16] was trained over 33,111 tiles at  $256 \times 256$  pixel level to distinguish Amyloid- $\beta$  in the form of diffuse plaques, cored plaques, and cerebral amyloid angiopathy (CAA).

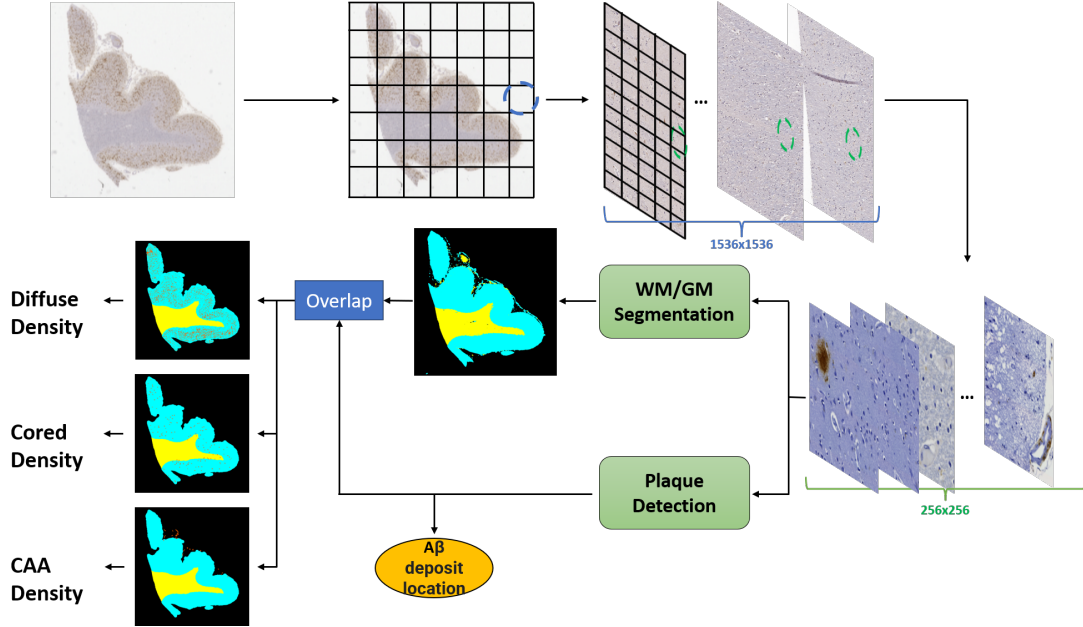


Figure 2.5. Overview of a pathology quantification model and visualization framework from Tang et al. [16] integrated with BrainSec.  $A\beta$  in the figure refers to Amyloid- $\beta$ .

### 2.3.3.1 Sliding window for inference

A sliding window method is applied to visualize the distribution and location of Amyloid- $\beta$  pathologies from a global view by generating WSI heatmaps of predictions. These heatmaps plot the location of each plaque predicted by  $f(\cdot)$  up to the full WSI view. A stride size of 16 pixels is used to walk through the WSI using the sliding window, resulting in the confidence heatmaps at a fraction of the resolution of original WSIs. This is helpful for reproducibility without excessive loss of information when implemented on modern devices equipped with Graphics Processing Units (GPUs). For each WSI, it will generate three confidence heatmaps corresponding to cored, diffuse, and CAA separately. After that, the cleaning and blob labeling is applied to the heatmaps. Subsequently, specific thresholds are applied to each plaque type, which converts the heatmaps to binary masks: probabilities below the threshold would be converted to zero, and above would be one. To study the distribution of different plaque types in GM and WM,  $g(\cdot)$  is incorporated into this framework to generate prediction heatmaps of GM and WM by using the same sliding window of stride size at 16 pixels to guarantee the output of  $f(\cdot)$  is of the same size of the output of  $g(\cdot)$  so that they can be pixel-wise overlapping. The heatmap of GM/WM will be displayed with the heatmap of plaques as shown in Figure 2.5: we visualize the distribution of each type of



---

**Algorithm 2** Automated Amyloid- $\beta$  pathology quantification in GM/WM regions

---

**Initialization:** the number of CAA, cored, and diffuse plaques in GM/WM as 0, store these six items into vector  $C^{1 \times 6}$ ; stride size  $w = 16$ , plaque CNN model  $f(\cdot)$ , BrainSec  $g(\cdot)$

**Input:** slides tiled into regions at  $1536 \times 1536$  pixels

**Output:**  $C^{1 \times 6}$

**for** each region at  $1536 \times 1536$  pixels **do**

the number of steps for running over the region  $T = 1 + (1536 - 256)/w$

**for**  $i=0$  to  $T$  **do**

**for**  $i=0$  to  $T$  **do**

extract one patch at  $256 \times 256$  pixels as  $P$

plaque prediction scores  $S = f(P)$

GM/WM prediction category  $R = g(P)$

based on  $S, R$ , get one-hot coding vector  $I^{1 \times 6}$

$C^{1 \times 6} = C^{1 \times 6} + I^{1 \times 6}$

**end for**

**end for**

**end for**

---

amyloid pathology, separately.

### 2.3.3.2 Plaque quantification

The plaque quantification algorithm is summarized in Algorithm 2. A zero vector  $P$  with the shape of  $1 \times 6$  is set up to record the number of each type of plaque (CAA, cored, diffuse) in GM and WM regions separately as the initialization. Then each WSI is normalized and tiled into  $1536 \times 1536$  pixel images. After that, the sliding window is applied to extract a patch  $P$  at  $256 \times 256$  pixels in an order from left to right, top to bottom. The patch will be the input of  $f(\cdot)$  and  $g(\cdot)$  to output the predictions in terms of both the type of deposit and the category of the region. These predictions are converted into a temporary one-hot vector  $I$ , added to  $C$  until the sliding window walks over the entire WSI. The final output is the vector  $C$  containing all quantification information of each type of plaques in GM and WM. All hyper-parameters involved in this component strictly follow the settings of the pathology detection model [16] as

our goal is to show the distribution of pathologies in GM and WM.

## 2.4 Experiments

### 2.4.1 Quantitative results

Since our goal is pixel-level GM/WM segmentation, we used three standard segmentation metrics — IoU score [38], DICE coefficient [129] and F1-score [130] to comprehensively compare segmentation masks from different methods.

Table 2.2: Pixel-wise IoU comparison for AD, NAD, and overall hold-out test set

|           | FCN              |                  | U-Net                              |                  | BrainSec                           |                                    | BrainSec+                          |
|-----------|------------------|------------------|------------------------------------|------------------|------------------------------------|------------------------------------|------------------------------------|
|           | <i>Model</i>     | <i>Post</i>      | <i>Model</i>                       | <i>Post</i>      | <i>Model</i>                       | <i>Post</i>                        |                                    |
| AD Back   | 75.46 $\pm$ 11.0 | 81.13 $\pm$ 9.17 | <b>97.32 <math>\pm</math> 0.61</b> | 96.80 $\pm$ 1.48 | 96.79 $\pm$ 0.67                   | 96.64 $\pm$ 1.51                   | 96.03 $\pm$ 2.31                   |
| AD GM     | 71.22 $\pm$ 11.7 | 76.07 $\pm$ 8.91 | 89.01 $\pm$ 4.92                   | 89.58 $\pm$ 5.12 | 91.10 $\pm$ 2.65                   | 91.84 $\pm$ 2.64                   | <b>92.28 <math>\pm</math> 2.44</b> |
| AD WM     | 54.69 $\pm$ 16.0 | 62.23 $\pm$ 14.0 | 79.42 $\pm$ 7.98                   | 82.53 $\pm$ 7.70 | 82.81 $\pm$ 4.44                   | <b>85.43 <math>\pm</math> 3.65</b> | 84.97 $\pm$ 4.55                   |
| AD Mean   | 67.12 $\pm$ 12.0 | 73.14 $\pm$ 9.66 | 88.58 $\pm$ 4.24                   | 89.64 $\pm$ 4.35 | 90.23 $\pm$ 2.29                   | <b>91.31 <math>\pm</math> 2.14</b> | 91.09 $\pm$ 2.91                   |
| AD F_W    | 70.48 $\pm$ 9.68 | 76.27 $\pm$ 7.03 | 90.91 $\pm$ 2.52                   | 91.44 $\pm$ 2.88 | 91.92 $\pm$ 2.32                   | <b>92.58 <math>\pm</math> 2.52</b> | 92.45 $\pm$ 2.74                   |
| NAD Back  | 78.63 $\pm$ 3.01 | 88.42 $\pm$ 1.55 | 97.15 $\pm$ 3.30                   | 97.36 $\pm$ 3.15 | <b>97.47 <math>\pm</math> 1.39</b> | 97.46 $\pm$ 1.98                   | 97.08 $\pm$ 1.42                   |
| NAD GM    | 72.47 $\pm$ 2.91 | 79.37 $\pm$ 2.95 | 93.75 $\pm$ 3.71                   | 94.42 $\pm$ 3.30 | 94.35 $\pm$ 1.97                   | <b>97.79 <math>\pm</math> 1.25</b> | 94.24 $\pm$ 1.85                   |
| NAD WM    | 40.70 $\pm$ 13.3 | 49.89 $\pm$ 12.8 | 78.68 $\pm$ 8.61                   | 81.25 $\pm$ 9.53 | 79.19 $\pm$ 6.69                   | 81.36 $\pm$ 6.94                   | <b>83.23 <math>\pm</math> 7.91</b> |
| NAD Mean  | 63.93 $\pm$ 3.54 | 72.56 $\pm$ 3.97 | 89.86 $\pm$ 3.15                   | 91.01 $\pm$ 3.36 | 90.34 $\pm$ 2.21                   | 91.34 $\pm$ 1.92                   | <b>91.52 <math>\pm</math> 2.15</b> |
| NAD F_W   | 73.05 $\pm$ 2.65 | 81.36 $\pm$ 3.97 | 94.31 $\pm$ 3.72                   | 94.98 $\pm$ 3.29 | 94.63 $\pm$ 2.40                   | <b>95.27 <math>\pm</math> 1.90</b> | 95.11 $\pm$ 2.20                   |
| Test Back | 76.73 $\pm$ 8.57 | 84.05 $\pm$ 9.17 | <b>97.25 <math>\pm</math> 1.96</b> | 97.02 $\pm$ 2.15 | 97.06 $\pm$ 1.01                   | 97.10 $\pm$ 1.46                   | 96.45 $\pm$ 1.95                   |
| Test GM   | 71.72 $\pm$ 8.93 | 77.39 $\pm$ 7.06 | 90.91 $\pm$ 4.90                   | 91.52 $\pm$ 4.94 | 92.40 $\pm$ 2.83                   | <b>93.06 <math>\pm</math> 2.71</b> | <b>93.06 <math>\pm</math> 2.20</b> |
| Test WM   | 49.09 $\pm$ 15.9 | 57.29 $\pm$ 14.3 | 79.12 $\pm$ 7.76                   | 82.02 $\pm$ 7.98 | 81.36 $\pm$ 5.42                   | 83.80 $\pm$ 5.28                   | <b>84.27 <math>\pm</math> 5.89</b> |
| Test Mean | 65.85 $\pm$ 9.33 | 72.91 $\pm$ 7.56 | 89.09 $\pm$ 3.71                   | 90.19 $\pm$ 3.84 | 90.27 $\pm$ 2.13                   | <b>91.32 <math>\pm</math> 1.94</b> | 91.26 $\pm$ 1.99                   |
| Test F_W  | 71.51 $\pm$ 7.58 | 78.31 $\pm$ 6.06 | 92.27 $\pm$ 3.35                   | 92.86 $\pm$ 3.40 | 93.00 $\pm$ 2.62                   | <b>93.65 <math>\pm</math> 2.58</b> | 93.55 $\pm$ 2.68                   |

*Model* is the inference output. *Post* is the inference output after post-processing. The results include: IoU scores and STD (Standard Deviation). AD is the average results on the 6 Alzheimer disease cases in test set. NAD is the average results on the 4 non-Alzheimer disease cases in test set. Test is the average results on all 10 WSIs. F\_W is the frequency weighted IoU for three categories.

**IoU score.** IoU score [38], as shown in (2.10) is to measure the overlapping amount between

the ground truth mask and the predicted mask.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}. \quad (2.10)$$

The slides shown in Figure 2.3 and Figure 2.7 clearly indicate the inherent heterogeneity of brain tissues among different WSIs, where GM/WM regions have a great variety of sizes and shapes. Considering its variety of shapes, we also select standard deviation (STD) to quantify how consistent and robust our methods are on different hold-out WSIs. Larger standard deviations indicate more performance inconsistency among heterogeneous slides. FCN achieves inferior performance compared to U-Net (its mean IoU is more than 20% lower than BrainSec and U-Net), which is similar to the results reported in [131, 132]. Hence, we will only compare BrainSec/BrainSec+ with U-Net in the subsequent sections.

From Table 2.2, BrainSec/BrainSec+ achieved the highest IoU and lowest standard deviation in the majority of indexes, which indicated that BrainSec/BrainSec+ was more stable and robust to a new hold-out test set compared to FCN and U-Net. Specifically, the mean IoU of BrainSec achieved more consistent segmentation results (less uncertainty) for both AD and NAD cases, since the difference between that of AD and that of NAD is only around 0.1% while there was more disparity in the confidence interval for AD vs. NAD cases for U-Net. The increased uncertainty in the segmentation results for AD compared to NAD is expected due to the existence of large amounts of plaques in AD cases.

Table 2.2 shows that BrainSec achieved 1.6% higher than U-Net in the mean IoU of AD cases, with smaller confidence intervals. Both BrainSec and U-Net achieved compatible IoU for NAD cases, but there was less uncertainty (smaller confidence interval) for BrainSec. Therefore, we can conclude that BrainSec shows stronger distinguishing ability in GM and WM among AD cases where boundaries may be more difficult to distinguish due to disease processes. BrainSec+ further improves BrainSec’s ability in WM among NAD cases and removes the need for post-processing.

**DICE coefficient.** Besides the IoU score [38], we selected DICE coefficient [129], another popular metric used in segmentation problems, to further evaluate the performance of BrainSec and BrainSec+. DICE coefficient [129], as shown in (2.11), is calculated by doubling the overlapping areas and dividing it by the total number of pixels in both areas.

Table 2.3: DICE coefficient comparison on the hold-out test set.

| Region | FCN [11] | U-Net [27] | BrainSec | BrainSec+ |
|--------|----------|------------|----------|-----------|
| GM     | 83.21    | 95.17      | 96.03    | 96.18     |
| WM     | 66.08    | 88.15      | 89.63    | 90.25     |

$$\text{DICE} = \frac{2 \times \text{Area of Overlap}}{\text{Total Sum of Pixels in All Areas}} \quad (2.11)$$

The results are summarized in Table 2.3. BrainSec/BrainSec+ was consistently superior to both FCN [11] and U-Net [27].

**F1-score.** In addition to the IoU score [38] and DICE coefficient [129], we also select F1-score as a third metric to compare the quantitative results. The results are summarized in Table 2.4. BrainSec/BrainSec+ can perform better performance, especially in WM, compared to FCN [11] and U-Net [27].

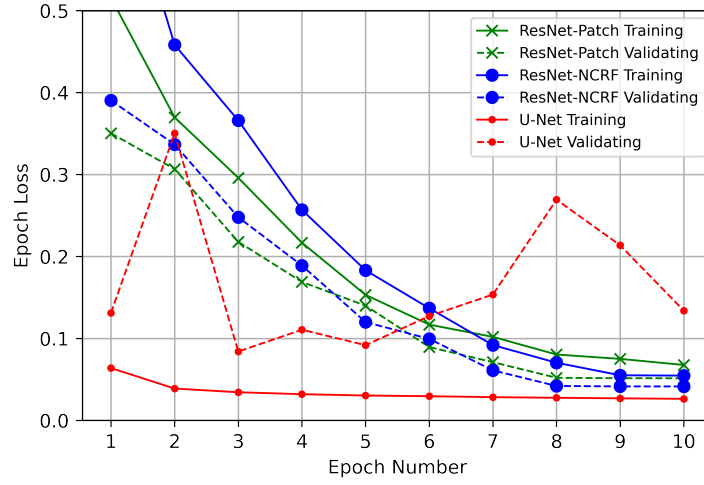
Table 2.4: F1-score comparison on the hold-out test set.

| Region | FCN [11] | U-Net [27] | BrainSec | BrainSec+ |
|--------|----------|------------|----------|-----------|
| GM     | 85.43    | 95.24      | 96.21    | 96.99     |
| WM     | 69.21    | 88.98      | 90.01    | 91.31     |

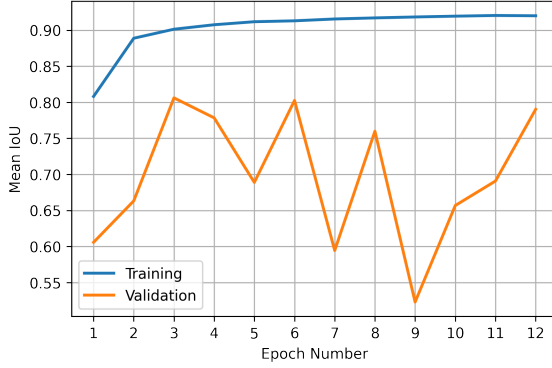
## 2.4.2 Training convergence of BrainSec

To compare the robustness of BrainSec/BrainSec+ with U-Net, we also analyzed the characteristics of training process of BrainSec/BrainSec+ and U-Net as shown in Figure 2.6. During our experiments, we found the epoch loss of U-Net on the validation set tended to oscillate with the epoch numbers while the loss on the training set strictly decreased (Figure 2.6a). However, the validation loss of BrainSec/BrainSec+ has a similar tendency compared to its training loss (both of them are strictly decreasing).

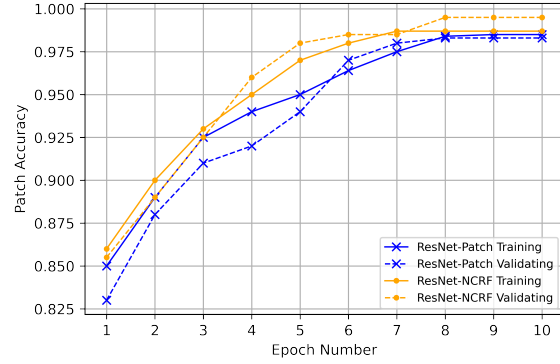
To further analyze the oscillation issue of U-Net and compare it with BrainSec/BrainSec+, we selected mean IoU of training and validating as performance metric as U-Net is a pixel-level architecture that outputs pixel-wise masks for GM and WM. Figure 2.6b shows its mean IoU



(a) Training/validating loss comparison



(b) U-Net mean IoU



(c) BrainSec/BrainSec+ patch accuracy

Figure 2.6. Training process comparison: visualization of the trends of training and validation comparing BrainSec/BrainSec+ with U-Net.

results on the validation set tend to oscillate across the number of trained epochs while the results on the training set strictly increase (Figure 2.6b). Besides, the mean IoU of training was over 0.9 while it could be 0.55 in the validation set at 9th epoch of training. Therefore, we performed five-fold cross-validation on the training and validation set to determine an optimal early stopping point for the number of training epochs. Then we trained the model for that number of epochs on the combined training-validation set and evaluated on the test set. The number of training epochs we obtained is 5 for U-Net.

On the other hand, BrainSec/BrainSec+ is a patch-level classification problem, so we chose patch-level accuracy to examine the training/validating process. Figure 2.6c shows the classifica-

tion accuracy of BrainSec/BrainSec+ continues to increase for both training and validation sets. Both achieve 0.96 accuracy or above after 6 epochs, indicating that their results are very close without any oscillations or instability.

### 2.4.3 Computation complexity

For our pathology quantification framework, we have two components. The pathology identification component [16] includes three major steps: tiling the slide into regions, deep learning model’s inference on these regions, and generating the heatmap for the entire slide. The advantage of BrainSec/BrainSec+ is that they can directly use the tile from the first step above, which saves data pre-processing time. As shown in Algorithm 2, we only needed to do tiling once for two models (pathology detection model and BrainSec/BrainSec+). However, FCN [98] and U-Net [27] require different data pre-processing steps, which can cause additional computation complexity for generating the distribution maps of each type of pathology. On the other hand, in terms of the computation complexity of the training and inference processes, FCN [98] and U-Net [27] require a much larger GPU memory ( 11.9 GB), but BrainSec/BrainSec+ only use around 700 MB. Hence, we believe BrainSec and BrainSec+ are more applicable on the edge where the devices do not have a large memory. Therefore, from this perspective, we conclude BrainSec and BrainSec+ are more deployable at the edge (where the hardware resources are limited) compared to FCN [98] and U-Net [27] when we need to integrate it into a pathology related task.

### 2.4.4 Segmentation visualization

Figure 2.7 shows the segmentation visualization of FCN, U-Net, BrainSec and BrainSec+ on hold-out AD and NAD cases separately. WSI-16 is a AD case while WSI-30 is a NAD case. In Figure 2.7, our methods segment the whole WSIs into three areas: GM, WM, and background, which are indicated by cyan, yellow, and black, respectively. The mask of FCN (Figure 2.7c, h) indicates FCN is not able to distinguish the tissue from background well (as shown in the top and bottom portions of the image where FCN detects GM when no tissue is present) while U-Net (Figure 2.7b, g), BrainSec (Figure 2.7d, i), and BrainSec+ (Figure 2.7e, j) can easily segment the tissue from background. The segmentation masks of U-Net, BrainSec, and BrainSec+ are

visually close to the ground-truth annotations generated by trained personnel.

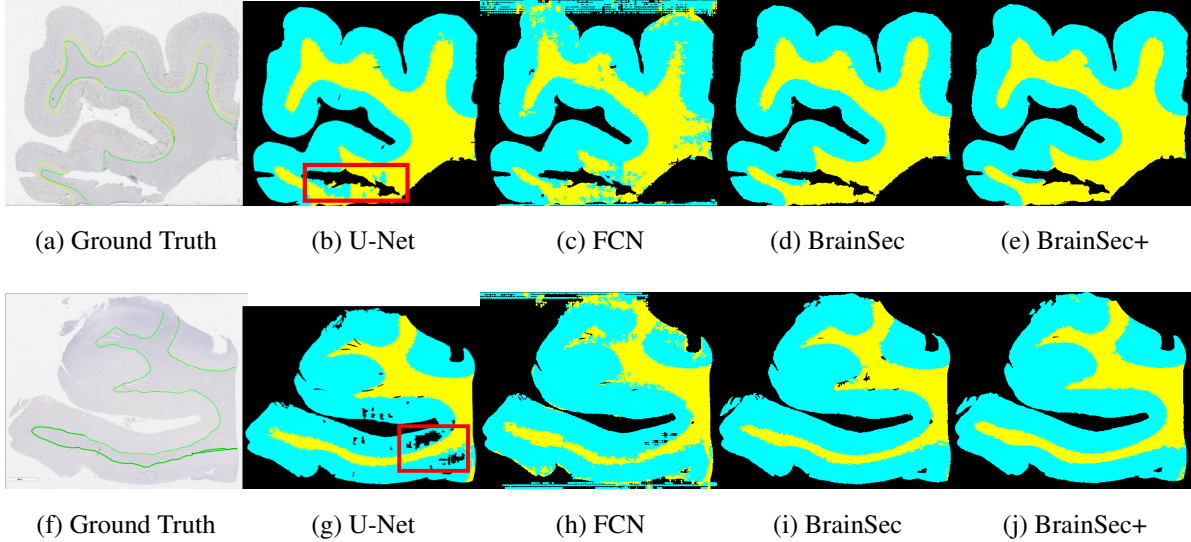


Figure 2.7. Segmentation masks visualization: WSI-16 is an AD case (top panels) and WSI-30 is a NAD case (bottom panels). GM, WM, and background are indicated by cyan, yellow, and black, respectively. The red box displays areas where U-Net struggled to correctly segment tissue. In the top WSI, U-Net has trouble differentiating between WM/GM. In the bottom WSI, U-Net has trouble segmenting tissue from the background.

### 2.4.5 BrainSec vs. BrainSec+

From the above sections, we show both BrainSec and BrainSec+ are able to produce consistent results. Although BrainSec requires less computation cost, its post-processing module refining the predicted masks still involves parameter tuning, which indicates we may need to tune this module manually on new datasets. To alleviate the need for such a module, we propose BrainSec+ to further automate the whole process by integrating a Neural Conditional Random Field layer into the neural networks. This additional layer could introduce computation costs but alleviate the need for a post-processing module. Hence, BrainSec+ has the potential to be deployed directly on other pathology tasks without further manual tuning. On the other hand, although BrainSec+ can eliminate the need for a post-processing module, the computation cost can potentially be a limitation. In our dataset, considering BrainSec+ only achieves trivial improvements in terms of performance but introduces heavy computation cost, we decided to use BrainSec in our Amyloid- $\beta$  plaque quantification framework. However, if the computational resources are ample and there is a need to deploy the framework to heterogeneous settings (e.g., slides from different brain

regions, scanners, or have different magnitudes) without manual post-processing, BrainSec+ will have the advantage.

#### 2.4.6 Pathology quantification with BrainSec

To investigate the compatibility of BrainSec with other pathology models, we integrated BrainSec into a recent plaque identification model [16] for quantitative pathological studies. Figure 2.8 shows one example of the output from the proposed framework: the distribution of each type of pathology can be visualized in GM and WM regions clearly. After we applied the proposed Algorithm 2, our framework detected 567 and 69 Amyloid- $\beta$  cored in GM and WM, respectively. It also detected 1358 and 114 Amyloid- $\beta$  diffuse in GM and WM, respectively. For this case, CAA was not detected. This is extremely advantageous for researchers to study the disease in a scalable way by automating this quantification process to relieve the manual quantification on gigapixel images. In Figure 2.8c, d, we can clearly observe that majority of pathologies appear in GM but also appear to a much lesser extent in WM, which is consistent with the findings in [10].

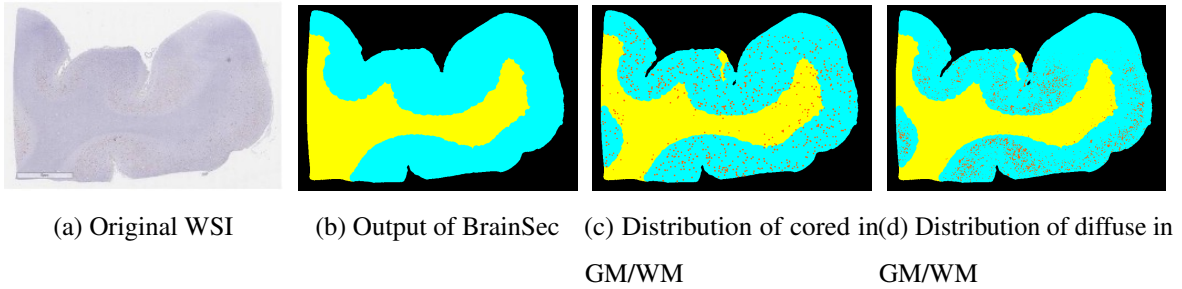


Figure 2.8. With the integration of BrainSec into the plaque identification model [16], the distribution of plaques (denoting in orange in c-d) in GM/WM can be visualized and quantified.

#### 2.4.7 Grad-CAM interpretation

**ResNet-Patch.** To investigate the differential morphologies corresponding to GM and WM separately, we used Gradient-weighted Class Activation Mapping (Grad-CAM) [28] to generate a coarse localization map where the relevant regions for predicting the concept are highlighted. Figure 2.9 provides visual explanations of what features BrainSec are deemed important for differentiating GM and WM for both AD and NAD cases, respectively. Grad-CAM [28] is meant for visual explanation of the deep learning models and provide a way for experts to check that



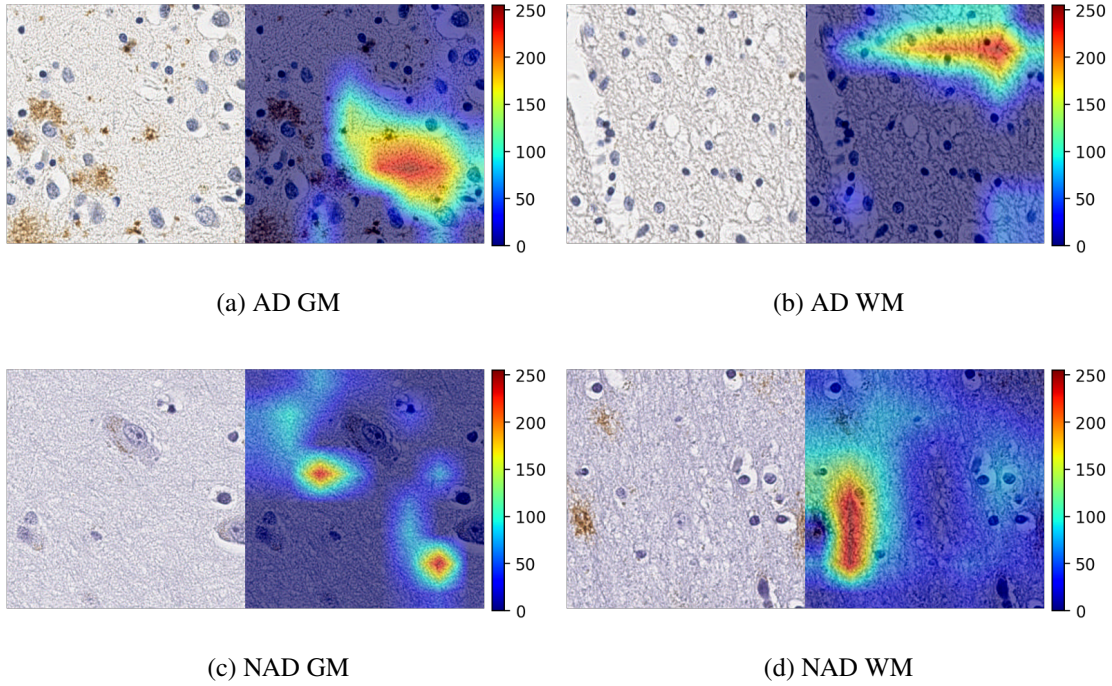


Figure 2.9. Grad-CAM on selected patches from both AD and NAD cases.

the model is learning from the right sets of features to perform the segmentation or classification tasks.

From Figure 2.9, we can see the energy around Amyloid- $\beta$  plaques (brown pixels in Figure 2.9a) and cells is relatively lower, while the highest energy is focusing on the textures of brain tissue. Based on pathology differences between GM and WM, one hypothesis is the BrainSec gravitates towards selecting areas that are composed of more randomly associated fibers (i.e. neuropil consisting of dendrites in GM, and more organized myelinated axons in WM). This is interesting as GM is also comprised of a more heterogeneous cell population (i.e. neurons and glial cells) while the predominate cell type in WM is oligodendrocytes. As a result, BrainSec not only seems to ignore pathological hallmarks of AD, such as Amyloid- $\beta$  plaques, but also other cellular components like neuronal and glial cells. Other abnormalities such as perivascular spaces, or artifacts are also avoided. The results have the potential to prove that our CNN feature extractor is relying on the differential textures from GM and WM rather than pathologies to determine the category of the patch, additional works with greater diversity (including other areas and stains) are needed. Therefore, our feature extractor is robust to datasets from both AD

cases and NAD cases, which is clinically reliable and reasonable.

#### 2.4.8 Evaluation on broader neuroanatomical areas

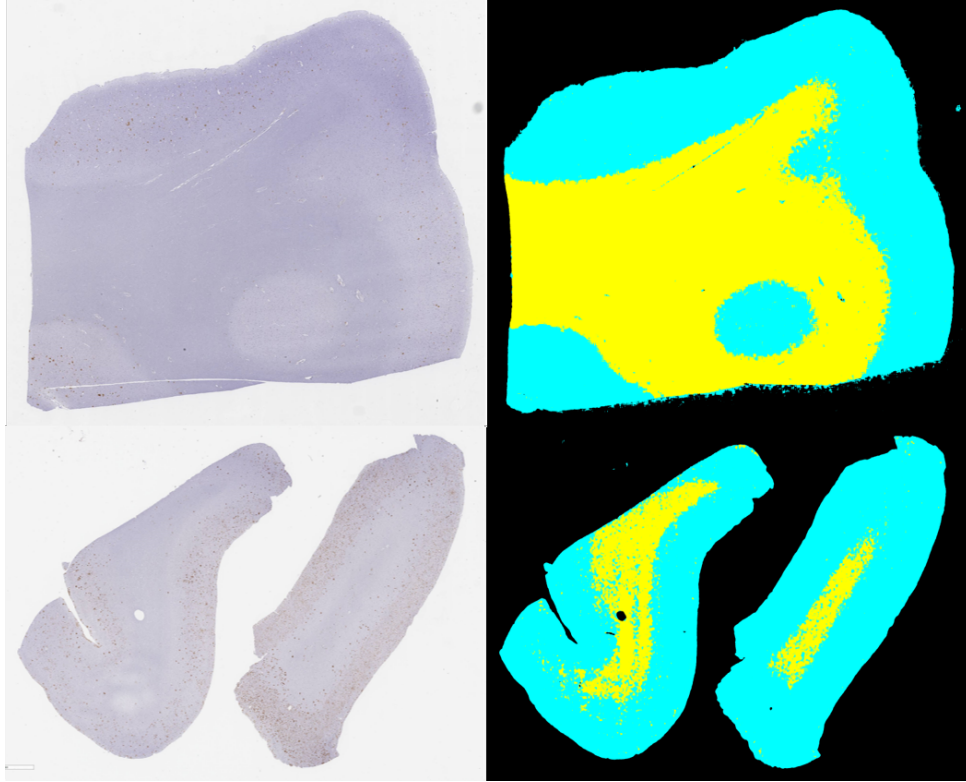


Figure 2.10. Results from running BrainSec on Frontal (top) and Parietal (bottom) WSIs.

Achieving good metrics on a medical image dataset through deep learning can sometimes be misleading. A successful model may display weak performance on similar tasks it was originally trained on if the setup or population is changed[133], which means testing BrainSec/BrainSec+ on a small dataset that contains only a single region of the brain would not guarantee us the model will perform well in all real-life scenarios. A convenience sample was prepared by selected 182 slides from three different brain regions: frontal, parietal and temporal from one institution (UC Davis), with one type of stain (4G8). This convenience sample also introduced 141 WSIs unseen by the model. Due to the large amount of data and the resource-intensive annotation process, no pixel-wise ground-truth was generated for the 182 WSIs. Meaning DICE scores and IoU metrics will not be available for this experiment. A successful prediction result would entail a well defined background/tissue boundary that visually matches the boundary observed

on the WSI, as well as not presenting any errors on WM/GM segmentation when seen from 20x magnification (two examples of successful prediction results can be seen on Figure 2.10). A table comparing the original dataset with the two datasets generated by the convenience samples can be seen on (see Table 2.1).

We invited two trained personnel to visually inspect the results for our model. The results of applying BrainSec/BrainSec+ to two new neuroanatomic areas (Parietal and Frontal) showed its tendency to display stable performance when facing heterogeneous scenarios. The results displayed WM/GM segmentation that would fit the expected areas from each class upon visual inspection (as seen in Figure 2.10), even with BrainSec/BrainSec+ never seeing any tissue from the Frontal or Parietal regions in training. We also tested BrainSec/BrainSec+ on 130 Temporal WSIs. Out of these slide, 122 WSIs - **94.23%** of the dataset - displayed WM/GM segmentation that accurately match the expected areas from each region after inspection.

## 2.5 Discussion

In this chapter, we propose an automated GM/WM segmentation pipeline, named BrainSec, for gigapixel pathology images. BrainSec consists of an interpretable CNN module and a post-processing module for reducing artifacts and residues existing in WSIs as well as generate XML annotations that are helpful for neuropathologists. The final output of XML annotations can be displayed on the original-size WSIs for neuropathological studies. We show BrainSec can consistently perform well on over more than 200 slides.

We demonstrate the easy compatibility of BrainSec into other pathology models to construct quantitative pathological studies. Specifically, we integrate it into a plaque detection model [16] and construct a framework to quantify the number and density of plaques in GM/WM regions. We show BrainSec is advantageous with quantitative pathological studies in a scalable way.

Currently, our study has the limitation of needing significant computing power and time to process each WSI. Using a NVIDIA Tesla T4 GPU, it takes us about 6 hours to generate the prediction maps for WM/GM and for the Amyloid- $\beta$  plaques for a gigapixel-level WSI.

Although the existing image processing methods may not require GPU support, they cannot achieve consistent performance on the hold-out test sets (as discussed in Section 2.1.1). Our

framework is more generalizable on the hold-out cases in real-world scenarios. GPUs are required for general deep learning methods, which could be a shortcoming. On the other hand, the advantages of deep learning methods include the generalizability of the model for a scalable analysis and consistent performance, as well as reduced manual labeling efforts. After we weigh these advantages and the shortcomings, we conclude that the deep learning framework holds a lot of potential for digital pathological analysis in the future.

In the future, we plan to implement BrainSec on many different setups to continue to test its generalizability. We aim to test BrainSec on different file formats used in WSI scanning and also aim to employ BrainSec on even more neuroanatomical areas. We also must investigate how much performance metrics can be improved if our training set contains more areas than just the Temporal region. Furthermore, the assessment of GM thickness in pathology images remains an area of ongoing research. Nonetheless, GM thickness constitutes a crucial parameter for tissue analysis. Consequently, our intention is to collaborate closely with neuropathologists to gain a comprehensive understanding of this metric's definition. Subsequently, we aim to develop and propose our approach for GM thickness measurement, supported by a substantiated conceptual framework.

# Chapter 3

## Enhanced Semi-supervised Learning to Improve Self-training

In Chapter 2, we prove the promising performance of different CNNs through supervised learning, which heavily relies on a large labeled dataset. For example, in Alzheimer disease (AD) research, neuropathologists usually assess histopathology by detecting and identifying extracellular Amyloid- $\beta$  plaques [134]. This is one of the hallmarks associated with AD [135, 136]. A recent work [16] claimed that more than 30,000 patches from gigapixel WSIs need to be annotated by neuropathology researchers to train a deep learning model to detect the Amyloid- $\beta$  plaques with acceptable accuracy. The labeling cost could be much higher in this case since medical experts must do the annotations. Hence the requirement of large amounts of curated data that involves substantial labeling cost may limit the wide adoption of supervised deep learning methods in real-world medical problems [31, 137].

Semi-supervised learning (SSL) is one that train models without requiring heavy annotations combining a small set of labeled samples with a large amount of unlabeled samples [30]. Consistency loss-based SSL methods involve both pseudo labels and data augmentation, showing their powerful performance on CIFAR-10 [17]. One drawback of these consistency loss-based SSL methods is that the imperfect class conditional distribution is used to generate pseudo labels and the over-reliance on pseudo labels make it difficult to correctly update the class conditional distribution [34]. In other words, incorrect pseudo labels during training may cause further

misclassifications in subsequent epochs, resulting in a feedback cycle of self-reinforcing errors that ultimately yields a poor-accuracy classifier [35]. A recent study [21] applied FixMatch [17] and MixMatch [30] to a histology dataset and showed that the performance of these state-of-the-art SSL methods are limited due to the above drawback.

### 3.1 Methodology

To deal with the above issue, inspired by [31] who claims that pre-training a classifier and then transferring it has the potential to outperform SSL in some settings (using 4000 labeled points from CIFAR-10), we design a novel two-stage SSL, SIM-FixMatch, to further reduce the labeling cost when labeled data is too rare for transfer learning. Our first stage is to employ self-supervised learning [37] for learning visual representations, which plays a similar role to pre-training in transfer learning but requires no labels. After the first stage, a pretrained encoder will be fed into the standard consistency loss-based SSL models. We employ our proposed scheme to segment GM and WM in Amyloid- $\beta$  stained histopathology images. To our best knowledge, our work is the first to tackle this task with minimal labeling cost and our proposed method outperforms FixMatch [17] when the amount of labeled data is much reduced (e.g., to 0.1% of total tiles from WSIs). In this section, we will introduce SIM-FixMatch, a two-stage SSL approach [36]. In the first stage, we utilize the self-supervised module to pretrain an encoder that learns the visual representations from unlabeled set. Then, we use this encoder as the input into a standard consistency loss-based SSL to leverage the information from both labeled and unlabeled set. Figure 3.1 shows the overall architecture.

#### 3.1.1 First stage: self-supervised pre-training

SimCLR [37] is a simple self-supervised framework for contrastive learning of visual representations on unlabeled images. As shown in Figure 3.1, an unlabeled image undergoes two random data augmentation operations  $t$  and  $t'$  and produces outputs  $h_i$  and  $h_j$  after going through the encoder network  $f(\cdot)$ .  $g(\cdot)$  is a projection head (multilayer perceptron with one hidden layer) to get  $z_i = g(h_i)$ .  $f(\cdot)$  and  $g(\cdot)$  are trained to maximize the agreement using the contrastive loss function

$$l_{i,j} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{I}_{[k \neq i]} \exp(\text{sim}(z_i, z_k)/\tau)},$$

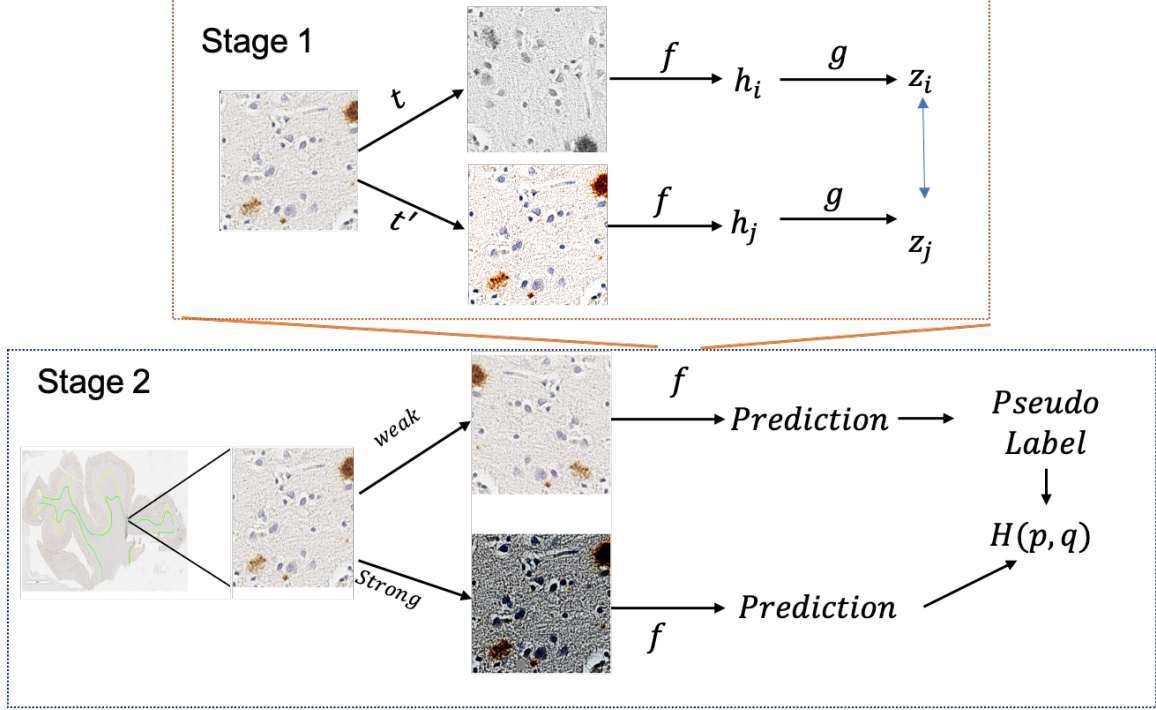


Figure 3.1. Two-stage Sim-FixMatch pipeline where encoder  $f$  in the 2nd stage is self-trained using the 1st stage.

where  $N$  is the size of a batch (two separate augmentation operators result in  $2N$  data points),  $\mathbb{I}_{[k \neq i]}$  is an indicator function, which is 1 if  $k \neq i$ ,  $\text{sim}(\cdot, \cdot)$  is cosine similarity, and  $\tau$  is a temperature parameter. The eventual goal for this stage is to train the encoder  $f(\cdot)$  for learning visual representations from unlabeled dataset. In our experiment, we use ResNet [55] for the encoder.

### 3.1.2 Second stage: standard FixMatch

In this study, we mainly adopt FixMatch [17], which combines pseudo-labeling and consistency regularization. Specifically, the pseudo label is generated based on a weakly-augmented unlabeled image (*weak*), which will be the target to compare with the output of the model on a strong-augmented version of the same unlabeled image (*strong*) as shown in Figure 3.1. As pseudo-labels generated here could be hurt by the imperfect class conditional distribution, we use the encoder  $f(\cdot)$  pretrained on unlabeled data from the first stage to provide an initialization for FixMatch. For the optimizer, instead of using standard SGD reported to have the best performance in [17], our experiments show Adam [138] performs better for our WSI dataset.

The “strong” augmentation operations include RandAugment and CTAugment [17] while the “weak” includes standard flip-and-shift augmentation.

## 3.2 Experiments

### 3.2.1 Data preparation

As downsampling [14] may lose medical features, we follow a patch-based method in [15] to divide WSIs into  $256 \times 256$  patches to cope with the ultra-high resolution. In this paper, 20 WSIs (12 AD cases and 8 NAD cases) were randomly selected for training and validation while the remaining 10 WSIs (6 AD cases and 4 NAD cases) were used for hold-out testing and inference. From the 20 WSIs, we selected one AD case and NAD case that have highest inter-rater agreement as the source of labeled patches while we kept the remaining 18 WSIs for generating unlabeled patches (around 600,000 patches). In our setting, we first generated 6,000 labeled patches (nearly 1% proportion of all patches) from 2 labeled WSIs, then we generated 6,000 labeled patches (nearly 0.1%) to further verify the effectiveness of our proposed method.

### 3.2.2 Effectiveness of the first stage

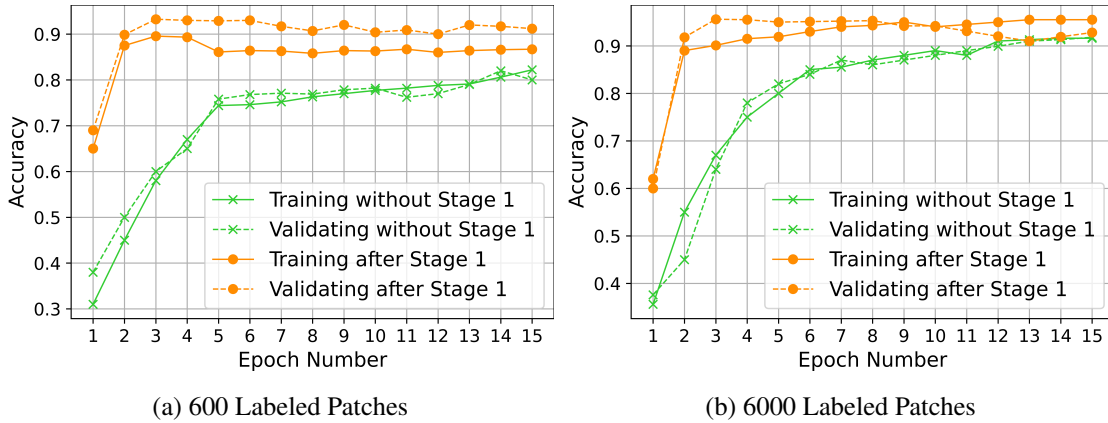


Figure 3.2. Trends of training and validation during the training process of FixMatch with or without the 1st stage.

To verify the effectiveness of our first stage, which is to learn visual representations on the unlabeled set and provide an encoder for the second stage, we visualized the training process of the second stage by using our method and baseline FixMatch. We trained both of them over 40 epochs and found that they converged quickly within 15 epochs. As shown in Figure3.2, with



the first stage, the model will be well-trained after only 3 epochs in the second stage, while it takes almost 20 epochs without the first stage. Besides, our proposed method starts from nearly 50% higher accuracy after the first epoch compared with original FixMatch, which shows the effectiveness of our proposed first stage on learning the representations via contrastive learning.

### 3.2.3 Numerical results

Table 3.1: Pixel-wise IoU Scores for AD, NAD, and overall test set

|           | 2 Labeled WSIs   |                  | 0.1% Labeled            |                         | 1% Labeled              |                         |
|-----------|------------------|------------------|-------------------------|-------------------------|-------------------------|-------------------------|
|           | FCN              | U-Net            | FixMatch                | <b>Proposed</b>         | FixMatch                | <b>Proposed</b>         |
| AD Back   | 61.04 $\pm$ 5.44 | 59.74 $\pm$ 13.9 | 93.15 $\pm$ 2.41        | <b>94.10</b> $\pm$ 2.21 | <b>96.59</b> $\pm$ 1.04 | 96.33 $\pm$ 1.05        |
| AD GM     | 46.98 $\pm$ 2.78 | 37.16 $\pm$ 9.93 | 78.57 $\pm$ 3.87        | <b>84.59</b> $\pm$ 3.27 | 87.12 $\pm$ 3.46        | <b>88.21</b> $\pm$ 3.84 |
| AD WM     | 27.75 $\pm$ 5.50 | 7.57 $\pm$ 6.02  | 56.66 $\pm$ 16.4        | <b>74.31</b> $\pm$ 3.36 | 73.94 $\pm$ 6.89        | <b>76.33</b> $\pm$ 4.77 |
| AD Mean   | 45.26 $\pm$ 3.55 | 35.40 $\pm$ 7.12 | 76.13 $\pm$ 5.89        | <b>84.34</b> $\pm$ 1.88 | 85.88 $\pm$ 3.09        | <b>86.95</b> $\pm$ 2.72 |
| NAD Back  | 66.66 $\pm$ 5.17 | 78.46 $\pm$ 18.5 | <b>97.07</b> $\pm$ 0.31 | 96.70 $\pm$ 0.73        | <b>97.71</b> $\pm$ 0.86 | 97.63 $\pm$ 0.88        |
| NAD GM    | 50.15 $\pm$ 0.49 | 59.59 $\pm$ 13.6 | 83.97 $\pm$ 7.76        | <b>86.58</b> $\pm$ 5.44 | 90.01 $\pm$ 4.02        | <b>90.93</b> $\pm$ 4.12 |
| NAD WM    | 19.72 $\pm$ 13.6 | 3.02 $\pm$ 3.09  | 22.72 $\pm$ 19.0        | <b>62.17</b> $\pm$ 7.04 | 59.71 $\pm$ 9.99        | <b>68.79</b> $\pm$ 7.22 |
| NAD Mean  | 45.51 $\pm$ 3.29 | 47.02 $\pm$ 10.9 | 67.92 $\pm$ 6.53        | <b>81.82</b> $\pm$ 2.15 | 82.47 $\pm$ 2.59        | <b>85.78</b> $\pm$ 2.00 |
| Test Back | 63.29 $\pm$ 5.81 | 68.28 $\pm$ 17.2 | 94.72 $\pm$ 2.71        | <b>95.14</b> $\pm$ 2.17 | <b>97.04</b> $\pm$ 1.08 | 96.85 $\pm$ 1.15        |
| Test GM   | 48.25 $\pm$ 2.66 | 46.13 $\pm$ 15.8 | 80.73 $\pm$ 6.01        | <b>85.39</b> $\pm$ 4.10 | 88.27 $\pm$ 3.78        | <b>89.30</b> $\pm$ 3.98 |
| Test WM   | 24.54 $\pm$ 9.80 | 5.75 $\pm$ 5.37  | 43.08 $\pm$ 24.0        | <b>69.45</b> $\pm$ 7.88 | 68.25 $\pm$ 10.7        | <b>73.31</b> $\pm$ 6.72 |
| Test Mean | 45.36 $\pm$ 3.26 | 40.05 $\pm$ 10.2 | 72.84 $\pm$ 7.18        | <b>83.33</b> $\pm$ 2.28 | 84.52 $\pm$ 3.26        | <b>86.48</b> $\pm$ 2.41 |

AD is the average results on the 6 Alzheimer’s disease cases in hold-out test set. NAD is the average results on the 4 non-Alzheimer’s disease cases in test set. Test is the average results on all 10 WSIs.

**IoU.** The results of IoU scores are summarized in Table 3.1. We selected the most updated version of FCN [11] and U-Net [27] as the supervised learning (SL) baselines for our comparison. Both of them are trained on only 2 labeled slides (1 AD case + 1 NAD case). Compared to the results reported in [26] that are trained on 20 labeled slides, their performance drastically deteriorates with reduced labeled WSIs from 20 to 2. The mean IoU scores for these two methods are only around 40%. FixMatch and our proposed method are trained on labeled patches (600 and 6000) from the same 2 labeled WSIs while the unlabeled patches are from other 18

unlabeled WSIs. For 6000 labeled patches setting, the labeled ratio is only 1%. FixMatch could achieve 84.52% of mean IoU while our proposed SIM-FixMatch is around 2% higher and has better performance in almost all classes, especially for the WM region in NAD cases (9.08% of improvement). Besides, our proposed method achieves 2.28% lower in terms of STD compared to original FixMatch. To further stress-test of our proposed method, we consider an extreme situation by using only 600 labeled patches (the labeled ratio is down to 0.1%). The improvement of SIM-FixMatch is significant, almost 40% of increase in terms of the WM region in NAD cases and 10.49% of increase in the mean IoU while the STD is still close to original FixMatch.

**DICE coefficient.** Besides IoU, we also use DICE coefficient [129] to further evaluate the proposed methods. When only 600 patches are labeled, the DICE coefficient of WM increases from 61.32% (FixMatch) to 82.67% (proposed). And when 6000 patches are labeled, it also gains 3% of improvement in WM.

### 3.2.4 Segmentation visualization

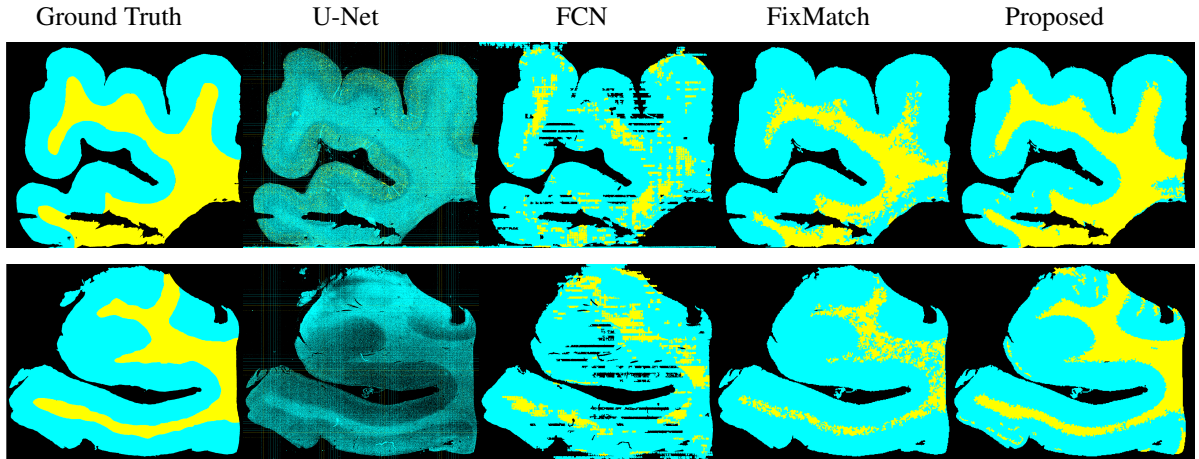


Figure 3.3. Segmentation mask visualization by only using 0.1% labeled patches from one AD case (top) and one NAD case (bottom). Both are from hold-out test set. Here GM, WM, and background are indicated by cyan, yellow, and black, respectively.

Figure 3.3 shows the segmentation visualization of SL methods (FCN [11], U-Net [27]) trained on the same 2 labeled WSIs and SSL methods (original FixMatch [17] and our proposed method) trained using 600 labeled patches from the same two labeled slides and unlabeled patches from the other 18 WSIs (the labeled ratio is only 0.1%). The masks of U-Net (Figure 3.3

the 2nd column) indicates that U-Net is unable to distinguish the WM from the GM; the masks of FCN (Figure 3.3 the 3rd column) have better visualization than U-Net but there are still many incorrectly labeled regions. FixMatch (Figure 3.3 the 4th column) is able to find the rough boundary between GM and WM but there are noisy pixels within WM, indicating it wrongly predicts some WM pixels as GM in the WM region. Our proposed method (Figure 3.3 the 5th column) could provide more distinguishable boundary for each region and the masks are the closest to the ground truth masks.

### 3.3 Discussion

In this chapter, we investigate the applicability of state-of-the-art semi-supervised learning in histology images and propose a two-stage approach to further improve the performance of SSL methods on Amyloid- $\beta$  stained WSIs at gigapixel level with the minimal labeling efforts. In our two-stage method, we verify the effectiveness of the first stage (self-supervised pretraining) by providing an encoder that has learned adequate visual representations among unlabeled data. Our proposed method outperformed the original FixMatch, especially in the case where labeled tiles are extremely rare (0.1%). While we showed promising results by running our experiments using randomly selected two WSIs, we will evaluate the selection criteria of WSIs more systematically in our future work.

These techniques have the potential to be applied to other classification and segmentation problems in medical images to minimize the expensive labeling cost. In addition, it takes nearly 3 days for SimCLR to train a good representation. Consequently, our future direction involves developing a task-based architecture to accelerate this process.

# Chapter 4

## Class-Imbalanced Semi-supervised Learning

Although deep neural networks have repeatedly achieved promising performance on various real-world tasks [55, 14, 139], their success heavily relies on the accessibility of large annotated datasets that typically involve labor-intensive labeling. The labeling cost could be even higher when specific domain knowledge such as in medical applications is required [139]. Semi-supervised learning (SSL) is one of the promising methods to reduce the labeling cost by leveraging unlabeled data. It has shown good results on standard image classification benchmarks using a very small labeled dataset [30, 17, 40]. To leverage unlabeled data effectively, the recent state-of-the-art SSL algorithms produce pseudo labels on unlabeled data, and then train the model by iteratively optimizing consistency loss as a regularization via these pseudo labels [140, 141].

These SSL methods [30, 17, 40, 140, 141] typically assume that labeled and unlabeled data are balanced (i.e., equal representation from different classes). One challenge for the wide-adoption of SSL in real-world tasks is that the performance may suffer from imbalanced training set [32, 142, 21], such as a long-tailed dataset from CIFAR-10 shown in Figure 4.1(a) [143]. Specifically, the model’s predictions can be biased towards the *majority classes* and away from *minority classes* that have few samples [144]. This affects the quality of pseudo labels, which could subsequently make the model even more biased. Consider the examples shown in Figure 4.1: using a popular SSL method FixMatch [17] on the long-tailed CIFAR-10 dataset, Figure 4.1(b)

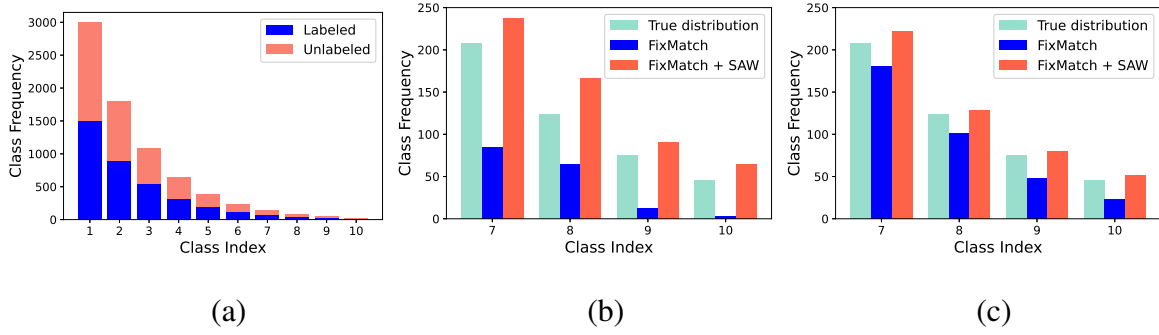


Figure 4.1. Experimental results on CIFAR-10 under the imbalanced ratio  $\gamma = 100$ . (a) Class distribution of labeled and unlabeled data. (b) Estimated distribution of the minority class after training 10 epochs. (c) Estimated distribution of the minority class after training 200 epochs.

shows that the numbers of pseudo labels for the minority classes 7-10 are significantly smaller than the true distribution after training for 10 epochs. In fact, the smallest class receives no pseudo labels at all. The number of minority classes’ pseudo labels increases but Figure 4.1(c) shows that they are still smaller than the true distribution after 200 epochs. The bias in the pseudo label distribution has prolonged the training and affected the classification performance.

A few recent studies [32, 145, 146, 33] try to tackle this problem, but all of these methods have made assumptions about the distribution of the unlabeled data. For example, CReST [33] assumes that unlabeled data have almost the same distribution as labeled data. DARP [32] utilizes the confusion matrix from labeled data to estimate unlabeled data distribution. These may not be applicable to real-world settings where the small labeled dataset is not representative of the population and there is no prior information about the unlabeled data, including its class distribution. This is the second challenge SSL faces for the wide-adoption in realistic tasks. Therefore, it remains an open problem to develop an effective class-imbalanced SSL algorithm without any assumptions about the distribution of unlabeled data.

To address the class-imbalanced issue, we assume labeled and unlabeled data share similar distribution [33] and focus on the first challenge of mitigating the impact of class imbalance on SSL. Based on the assumption, we show that using smoothed weighting on the consistency loss is effective in countering the class imbalance problem. We then turn to the second challenge of unknown distribution of unlabeled data and propose Smoothed Adaptive Weighting (SAW) that can automatically estimate the class distribution of unlabeled data and assign weights to

the consistency loss per epoch. As shown in Figure 4.1(b) and (c), SAW forces the model to have a rough awareness of class distribution in unlabeled data, and adjust the weights for all classes accordingly based on the model’s awareness. More biased pseudo distribution will trigger a stronger response of weights derived from the model’s self-awareness. This self-adjusting scheme could gradually refine the pseudo labels to move closer to the true distribution of the unlabeled data [39].

## 4.1 Related work

**Semi-supervised learning.** Semi-supervised learning (SSL) is a learning method that aims to improve the performance of supervised learning by leveraging unlabeled data. One commonly-used SSL approach is to rely on pseudo-labeling, where a model trained with labeled data and possibly unlabeled data with previously generated pseudo-labels is used to generate pseudo labels for each unlabeled sample [30, 17, 40, 141, 147]. These pseudo labels are then combined with labeled data to train the final model. Temperature scaling is applied to generate a soft pseudo label based on the model’s predictive probability [30, 40, 148, 149]. To promote consistency across predictions over unlabeled samples, consistency regularization can be applied across different augmented views of the same unlabeled training samples [140, 150, 151]. Recent state-of-the-art SSL algorithms such as FixMatch [17] and ReMixMatch [40] combine both pseudo labels and consistency regularization. However, all of these aforementioned methods assume balanced class distribution in both labeled and unlabeled data and may face performance degradation in real-world imbalanced datasets [142, 21].

**Imbalanced supervised learning.** Real-world datasets are typically of “long-tailed” distribution [152, 153, 154, 155, 156]. Supervised learning on imbalanced distribution has been widely studied. Popular methods to re-balance the class distributions include re-sampling the data [157, 158, 159] and re-weighting the loss function [160, 161, 162, 163]. However, simple re-balancing may lead to over-fitting to the minority classes. To address this issue, [164] proposed “effective number of samples” to measure the real volume of each class. Another direction is to transfer and distill the knowledge from majority classes to minority [152, 155, 165]. Although these methods perform effectively on supervised learning, they require labels for all classes in

the entire dataset while labeled dataset is small in SSL framework. Therefore, the applicability of these methods to SSL is unclear.

**Imbalanced semi-supervised learning.** A recent study shows that class-imbalance problem can be greatly benefited from unlabeled data, sometimes even more so than labeled data, by either using pseudo-labeling in SSL or without any labels in self-supervised learning [145]. Similarly, [33] proposes a class-rebalancing self-training framework named CReST to expand the labeled data by adding pseudo-labeled samples. One obvious drawback of these approaches is that they ignore the impact on the quality of the pseudo labels due to the difference in class distributions between the labeled and unlabeled data: different distributions may bias the model and generate biased pseudo labels. DARP [32] introduces distribution aligning refinery of pseudo-label to softly refine the pseudo labels from the model. However, DARP also assumes that labeled and unlabeled data can be sampled from similar input distribution: they approximate the confusion matrix of unlabeled data by directly using that of labeled data. Another limitation of these methods is that they only report the results on a balanced test set [32, 33]. The effectiveness of their algorithms need to be tested against realistic class-imbalanced scenarios where training and testing set may have totally different distributions.

## 4.2 Methodology

### 4.2.1 Preliminaries

We first give a brief review on semi-supervised learning (SSL). Consider a classification problem with  $C$  classes. An input vector and corresponding one-hot label are denoted  $x \in \mathbb{R}^d$  and  $y \in \{0, 1\}^C$ , respectively. Here  $d$  is the dimension of the input. The training data contains  $m$  labeled instances  $\mathcal{D}^{(l)} = \left\{ (x_i^{(l)}, y_i^{(l)}) \right\}_{i=1}^m$  and  $n$  unlabeled instances  $\mathcal{D}^{(u)} = \left\{ x_j^{(u)} \right\}_{j=1}^n$ , with  $m \ll n$ . The number of training labeled instances in  $\mathcal{D}^{(l)}$  of class  $k$  is denoted as  $m_k$ , i.e.,  $\sum_{k=1}^C m_k = m$ . Similarly, we have  $n_k$  for unlabeled instances in  $\mathcal{D}^{(u)}$  of class  $k$  with  $\sum_{k=1}^C n_k = n$ . In general, the distribution of unlabeled data  $\{n_k\}_{k=1}^C$  is unknown. Without loss of generality, we assume  $\{m_k\}$  and  $\{n_k\}$  are sorted by cardinality in descending order, i.e.,  $m_1 \geq m_2 \geq \dots \geq m_C$ , and  $n_1 \geq n_2 \geq \dots \geq n_C$ . In a class-imbalanced dataset, the class distribution is skewed, i.e.,  $m_1 \gg m_C$  or  $n_1 \gg n_C$ . Here we define  $\gamma_l = \frac{m_1}{m_C}$  and  $\gamma_u = \frac{n_1}{n_C}$  to

measure the degree of class imbalance in labeled and unlabeled data, respectively.

The goal of SSL is to train a classifier  $h(x; \theta) : \mathbb{R}^d \rightarrow [0, 1]^K$  using the above training data. Here  $h(x; \theta)$  is parameterized by  $\theta \in \Theta$  and its  $k$ -component  $h(x; \theta)_k \in [0, 1]$  is the predictive probability for the  $k$ -th class given an input  $x$ . The training process is to minimize the generalization loss in both labeled and unlabeled data

$$\min_{\theta \in \Theta} \sum_{i=1}^m \mathcal{L}_l(x_i^{(1)}, y_i^{(1)}; \theta) + \sum_{j=1}^n \Omega(x_j^{(u)}; \theta), \quad (4.1)$$

where  $\mathcal{L}_l$  is the per-sample supervised loss (e.g., cross-entropy) for labeled data and  $\Omega$  is the per-sample regularization for unlabeled data. Note that some SSL algorithms (e.g., FixMatch [17]) include both labeled and unlabeled data in the regularization term, where we can simply replace the second term of (4.1) by  $\sum_{j=1}^{n+m} \Omega(x_j; \theta)$ , for  $x_j \in \mathcal{D}^{(1)} \cup \mathcal{D}^{(u)}$ . One of the widely used regularization is consistency loss [140, 17, 151]. This kind of regularization forces SSL to predict consistency on different views of the same sample. To utilize  $\mathcal{D}^{(u)}$  effectively, many SSL algorithms infer their labels by some pseudo labeling scheme  $p$  defined as  $\left\{ p(x_j^{(u)}; \theta) \in [0, 1]^C \mid \sum_{k=1}^C p(x_j^{(u)}; \theta)_k = 1 \right\}_{j=1}^n$ . Then, we can formulate the consistency loss as follow:

$$\mathcal{L}_c(x; \theta) := \sum_{k=1}^C p(x; \theta)_k \cdot \log(h(\text{pertub}(x); \theta)_k). \quad (4.2)$$

Here  $\text{pertub}(x)$  defines the specific augmentation scheme. In VAT [151], this operator is an adversarial perturbation that disregards the input class distribution. FixMatch [17] adopts strong and weak augmentation as perturbations in  $\text{pertub}(x)$  and  $p(x; \theta)$ , respectively. Note that other consistency loss like the mean-square error can also be used:  $\|p(x; \theta) - h(\text{pertub}(x); \theta)\|_2^2$ . For simplicity, in the following content, we will only consider consistency loss (4.2) as the regularization term, i.e.,  $\Omega = \mathcal{L}_c$ .

## 4.2.2 A closer look at the weight in consistency loss

In this subsection, we investigate the impact of weighted consistency loss for class-imbalanced SSL. As the imbalanced class distribution tilts towards the majority classes, previous works in supervised learning [160, 161, 162, 163] realign the distribution by different re-weighting strategies. In SSL, efforts on re-weighting schemes remain limited. Here we define a weighting



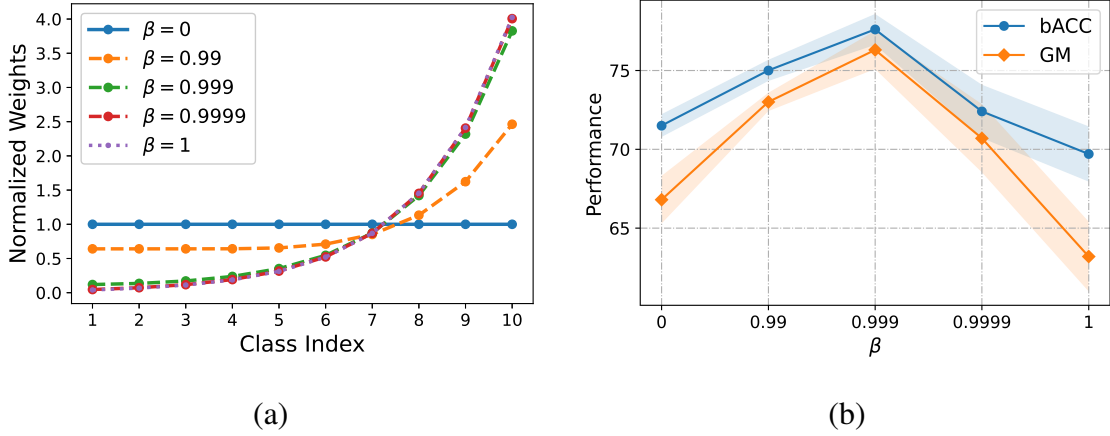


Figure 4.2. Empirical illustration of the performance impact of weighting functions with different  $\beta$  using CIFAR10-LT under the imbalance ratio  $\gamma = 100$ . (a) The normalized weights with different values of  $\beta$ . (b) Accuracy performance measured bACC and GM (with standard deviation) of FixMatch [17] under class-rebalanced weighting scheme [166] with different  $\beta$ .

consistency loss in the cross entropy format as:

$$\mathcal{L}_{cw}(x; w, \theta) := \sum_{k=1}^C w_k \cdot p(x; \theta)_k \cdot \log(h(\text{pertub}(x); \theta)_k), \quad (4.3)$$

where  $w_k$  is the weight for the  $k$ -th class. We use the standard cross-entropy loss with uniform weights as  $\mathcal{L}_l$ . A long-tailed version of the CIFAR10 with a class-imbalanced ratio  $\gamma := \gamma_l = \gamma_u = 100$  is studied. We test FixMatch [17] with its consistency loss replaced by  $\mathcal{L}_{cw}$ . Following the class re-balanced weighting scheme in [164], we set the weight to be inversely proportional to the effective number of samples for each class, i.e.,

$$w_k \propto 1/E_k, \text{ where } E_k = (1 - \beta^{n_k})/(1 - \beta). \quad (4.4)$$

We normalize  $w_k$  such that  $\sum_{k=1}^C w_k = C$ . The degree of smoothing is determined by the value of  $\beta$  as  $\beta = 0$  leads to uniform weights. When  $\beta$  goes to 1,  $E_k \rightarrow n_k$ , which corresponds to the weight by strictly inverse class frequency. The normalized weights with different values of  $\beta$  are shown in Figure 4.2(a). The performance is evaluated by balanced accuracy (bACC) [156, 167] and geometric mean scores (GM) [168, 169]. In Figure 4.2, we observe that selecting weights by either strictly inverse class frequency or uniform weight produce the worst results while smoothed versions of weights show some improvements. This observation indicates the importance of the smoothness for the weight in consistency loss. Additionally, the value of  $\beta$  in [164] corresponds

to  $(N - 1)/N$ , where  $N$  is a hyper-parameter representing the volume of each class. An interesting observation is that the optimal value of  $\beta$  in our experiments corresponds to the  $N$  being the average of  $\{m_k + n_k\}$  over all  $k$ 's.

### 4.2.3 SAW: Smoothed adaptive weighting scheme

**Smoothed weighting.** Based on the above observation, we propose a smoothed weighting scheme for SSL. We first start with the same case that the unlabeled data  $\mathcal{D}^{(u)}$  shares the same class distribution as the labeled data  $\mathcal{D}^{(l)}$ , i.e.  $\frac{n_1}{m_1} = \frac{n_2}{m_2} = \dots = \frac{n_C}{m_C}$ . Therefore, we can keep the same weight in consistency loss and supervised loss, i.e.,  $w^{(l)} = w^{(u)}$ . The loss function becomes  $\min_{\theta \in \Theta} \sum_{i=1}^m \mathcal{L}_{lw}(x_i^{(l)}, y_i^{(l)}; w, \theta) + \sum_{j=1}^n \mathcal{L}_{cw}(x_j^{(u)}; w, \theta)$ , where  $\mathcal{L}_{lw}(x, y; w, \theta) = \sum_{k=1}^C w_k \cdot y_k \cdot \log(h(x; \theta)_k)$  is a weighted supervised loss and  $\mathcal{L}_{cw}$  is defined in (4.3). Many re-weighting schemes for supervised learning add weights outside each per-sample loss while our weight  $w$  is inside the loss. These two are equivalent for  $\mathcal{L}_{lw}$  as there is only one non-zero term. However, for  $\mathcal{L}_{cw}$ , when the pseudo label is not a one-hot vector, we cannot add the weight outside the per-sample consistency loss.

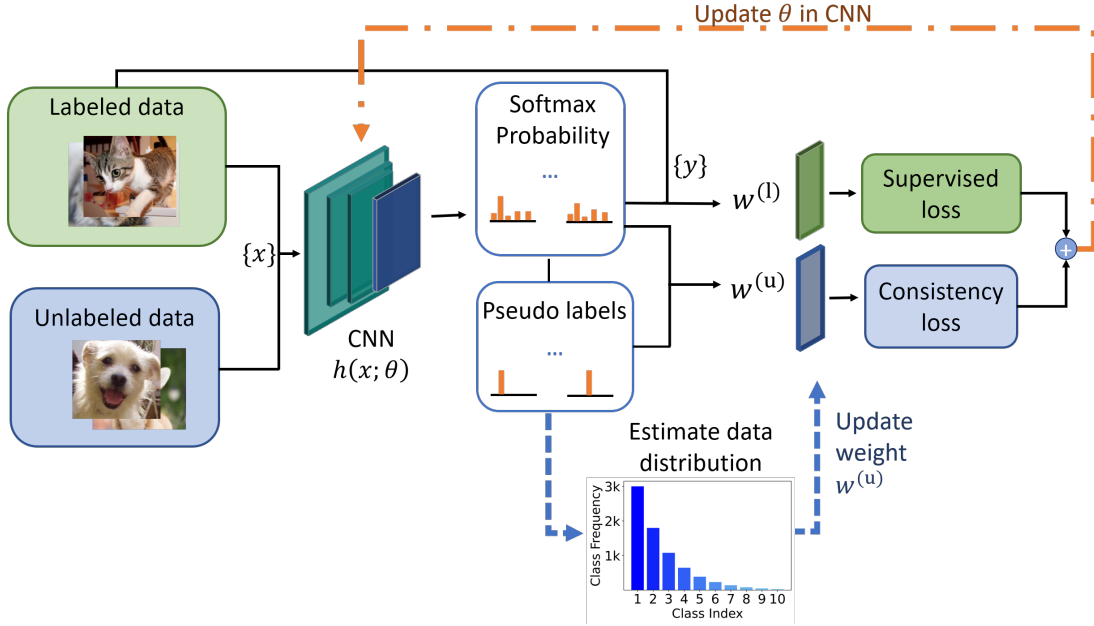


Figure 4.3. Illustration of the SAW framework.

**Adaptive weighting.** In real-world scenarios, the class distribution in the unlabeled data is unknown. The weight in consistency loss therefore cannot be directly inferred from labeled data.

In [32], the distribution is estimated from the confusion matrix of a subset of labeled dataset. This estimation is limited as it assumes that confusion matrices of labeled and unlabeled datasets are similar. Besides, when labeled data is scarce, the confusion matrix of a subset of labeled dataset may fail to represent the whole dataset. Alternatively, the pseudo labels can be used to estimate the class distributions of the unlabeled data. However, Figure 4.1 shows that the distribution of the minority classes obtained from pseudo labels by FixMatch is significantly underestimated, especially in the first few epochs. Thus, our aim is to re-balance this pseudo labels' distribution by using them to adaptively adjust a smoothed weight term  $w^{(u)}$  for consistency loss, which in turn leads to an improved estimation of the pseudo labels in the next epoch. In general, we utilize gradient descent methods or its variants, such as Adam [138], to update the  $\theta$  in the training process, i.e.,  $\theta \leftarrow \theta - \eta \cdot \nabla \mathcal{L}(\theta, w^{(u)})$ , where  $\eta$  is the learning rate and  $\mathcal{L}$  is the loss function:  $\mathcal{L}(\theta, w^{(u)}) = \sum_{i=1}^m \mathcal{L}_{lw}(x_i^{(1)}, y_i^{(1)}; w^{(1)}, \theta) + \sum_{j=1}^n \mathcal{L}_{cw}(x_j^{(u)}; w^{(u)}, \theta)$ . The weight  $w^{(1)}$  is fixed based on the distribution of the labeled data, while the weight  $w^{(u)}$  is initialized to be uniform. After training for one epoch, we compute the one-hot pseudo label  $p(x, \theta)$  via the updated  $\theta$ . Then class distributions can be estimated by  $n_k = \sum_{j=1}^n p(x_j, \theta)_k$ , for  $k = 1, \dots, C$ . As motivated in Section 4.2.2, the weight should be proportional to  $(1 - \beta)/(1 - \beta^{n_k})$ , but singularity may rise when estimated  $n_k = 0$ . Here we adjust the  $n_k$  by  $\hat{n}_k = \max(n_k, 1)$ . Following the observation in Section 4.2.2, we update  $\beta = (N - 1)/N$ , where  $N = \sum_{k=1}^C (m_k + \hat{n}_k)/C$ . With the normalization, the weight  $w^{(u)}$  is updated as

$$w_k^{(u)} = C / \sum_{t=1}^C \frac{1 - \beta^{\hat{n}_k}}{1 - \beta^{\hat{n}_t}}, \text{ for } k = 1, \dots, C. \quad (4.5)$$

During the training process, improved quality of pseudo labels could result in a better estimation of class distributions. We call this proposed algorithm as smoothed adaptive weighting (SAW), which is summarized in Algorithm 3 and Figure 4.3.

#### 4.2.4 SaR: Self-adaptive refinement on soft labels

We first set up the problem in class-imbalanced semi-supervised learning (SSL). In an  $C$ -class classification task, an input vector and its corresponding one-hot label are denoted as  $x \in \mathbb{R}^d$  and  $y \in \{0, 1\}^C$ , respectively. Here  $d$  is the dimension of the input. There are a labeled set  $\mathcal{D}_1 = \{(x_i^{(1)}, y_i^{(1)})\}_{i=1}^m$  and an unlabeled set  $\mathcal{D}_u = \{x_j^{(u)}\}_{j=1}^n$ , with  $m \ll n$ . Let  $m_k$  be the

---

**Algorithm 3** The SAW Framework

---

**Data:** Labeled data  $\{(x_i, y_i)\}_{i=1}^m$ , unlabeled data  $\{x_j\}_{j=1}^n$ , weight in supervised loss  $w^{(1)}$ , number of classes  $C$ , learning rate  $\eta$ , max epoch  $T$

Initialize uniform weights in consistency loss  $w^{(u)}$  and model parameter  $\theta$

**while**  $t < T$  **do**

**for**  $1, \dots, K$  **do**

    Sample batches label data  $\{x, y\}$

    Sample batches unlabeled data  $\{x\}$

    Update  $\theta \leftarrow \theta - \eta \cdot \nabla \mathcal{L}(\theta, w^{(u)})$

**end for**

  Compute pseudo labels  $p(x; \theta)$  and estimate distribution  $(n_1, n_2, \dots, n_C)$

  Adjust the distribution  $(\hat{n}_1, \hat{n}_2, \dots, \hat{n}_C)$

  Update weight  $w^{(u)}$  based on (4.5)

**end while**

**Return:**  $\theta$

---

number of training labeled instances in  $\mathcal{D}_l$  of class  $k$ , i.e.,  $\sum_{k=1}^C m_k = m$ . Similarly,  $n_k$  is for unlabeled instances in  $\mathcal{D}_u$  of class  $k$  with  $\sum_{k=1}^C n_k = n$  where  $n_k$  is *unknown*. Without loss of generality, we assume  $\{m_k\}$  and  $\{n_k\}$  are sorted by cardinality in descending order, i.e.,  $m_1 \geq m_2 \geq \dots \geq m_C$ , and  $n_1 \geq n_2 \geq \dots \geq n_C$ . Here the skewed class distribution is considered, i.e.,  $m_1 \gg m_C$  or  $n_1 \gg n_C$ . The fraction  $\gamma_l = \frac{m_1}{m_C}$  and  $\gamma_u = \frac{n_1}{n_C}$  measure the degree of class imbalance in labeled and unlabeled data, respectively.

SSL aims to train a classifier  $h(x; \theta) : \mathbb{R}^d \rightarrow [0, 1]^C$  using the above training data. Here the  $k$ -component  $h(x; \theta)_k \in [0, 1]$  is the predictive probability for the  $k$ -th class given an input  $x$ . To utilize  $\mathcal{D}_u$  effectively, many state-of-the-art SSL algorithms infer their labels using pseudo labeling schemes denoted by

$$\left\{ \hat{y}^{(u)} \in [0, 1]^C \mid \sum_{k=1}^C \hat{y}^{(u)}(k) = 1 \right\}_{j=1}^n.$$

Pseudo labels are generated based on the classifier  $h(x; \theta)$  with a stochastic data augmentation

function  $\alpha$ . Specifically, one can obtain a one-hot vector as pseudo label by

$$\hat{y}^{(u)}(k) = \delta(h(\alpha(x^{(u)}); \theta)_k), k = 1, \dots, C, \quad (4.6)$$

where  $\delta$  is an operator to transform a soft label  $h(\alpha(x^{(u)}); \theta)$  into a one-hot vector. With these pseudo labels, the training process is to minimize the loss in both labeled and unlabeled data

$$\min_{\theta \in \Theta} \sum_{i=1}^m \mathcal{L}(h(x_i^{(1)}; \theta), y_i^{(1)}) + \sum_{j=1}^n \hat{\mathcal{L}}(h(\hat{\alpha}(x_j^{(u)}); \theta), \hat{y}_j^{(u)}), \quad (4.7)$$

where  $\mathcal{L}$  and  $\hat{\mathcal{L}}$  are the per-sample losses (e.g., cross-entropy). The first term works as a supervised loss function for labeled data while the latter one is a regularization term. Note that some SSL algorithms use different distance-based loss function for labeled and unlabeled data. One of the widely used regularization is consistency loss [140, 17, 151]. This type of regularization forces SSL to predict consistency on different views of the same sample. Here  $\hat{\alpha}$  refers to another augmentation different from  $\alpha$  in (4.6). For simplicity, in the following content, we will only consider consistency loss as the regularization term.

**Self-adaptive refinement on soft labels** Although traditional reweighting [170, 164, 160, 161, 171] and resampling [172, 173, 157, 174] are popular techniques dealing with imbalanced datasets and show their effectiveness on supervised learning (SL), they cannot be directly applicable to SSL since the majority of samples are unlabeled. Instead, we focus on the threshold-moving method [175] to improve the quality of pseudo labels in class-imbalanced SSL. One drawback of consistency loss-based SSL methods is that the imperfect class distribution is used to generate pseudo labels and the over-reliance on pseudo labels makes it difficult to correctly update the class distribution. It motivates us to take one step back and focus on refining the soft labels and adjusting the thresholds for pseudo label generation. Threshold-moving is a classical method for the binary-classification tasks. It has been extended to multi-class problems for supervised learning. For example, Zhou et al. [175] adjusted the output of neural network to move the output threshold toward classes with higher misclassification costs. This is very different from re-sampling [172, 173, 158, 159, 174] as it relies on the manipulation of the outputs of the classifier instead of re-balancing training data. Threshold-moving is more applicable to SSL than traditional re-sampling or re-weighting schemes since it does not rely on the pre-defined distribution of the unlabeled data.

Here we refine the pseudo labels (4.6) as

$$\hat{y}^{(u)}(k) = \delta(s_k \cdot h(\alpha(x^{(u)}); \theta)_k), k = 1, \dots, C. \quad (4.8)$$

where  $s = [s_k, k = 1, \dots, C]$  is a positive scaling vector to mitigate the confirmation bias [141]. We refer to this vector as *Mitigating Vector* and will describe how it is calculated in the next subsection. Note that some SSL algorithms directly use soft labels as pseudo labels. In that circumstance, (4.8) becomes

$$\hat{y}^{(u)}(k) = \frac{s_k \cdot h(\alpha(x^{(u)}); \theta)_k}{\sum_{j=1}^C s_j \cdot h(\alpha(x^{(u)}); \theta)_j}, \quad (4.9)$$

with the normalization.

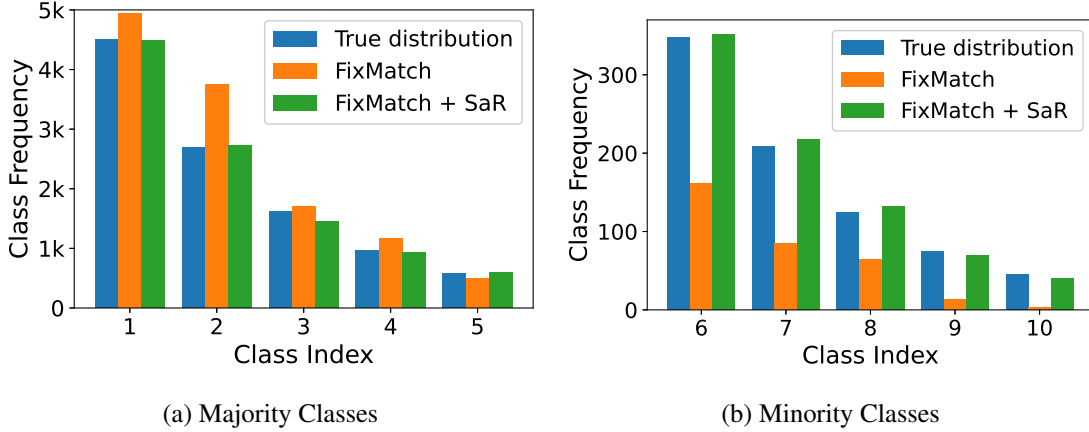


Figure 4.4. Confirmation bias on CIFAR10-LT under imbalanced ratio  $\gamma = 100$ : (a) the estimated distribution of majority classes; (b) the estimated distribution of minority classes.

**Mitigating Vector.** Inspired by the cost-sensitive classification [175], we turn our attention to the learning effect of unlabeled data during the training process. Consider a long-tailed version of the CIFAR-10 with the imbalance ratio (the ratio between the number of samples of the most frequent class and that of the least frequent class) as  $\gamma := \gamma_l = \gamma_u = 100$ . In the same figure, we see that pseudo labels generated from FixMatch [17] are highly biased towards the majority classes. The original imbalance ratio can be further accentuated from 100 to 1000 [32]. To address this issue, we propose Self-adaptive Refinement (SaR) to take the confirmation bias of each class in SSL into account. Specifically, we adjust the soft labels so as to re-balance the contributions of samples from both the majority and the minority classes. To achieve this,

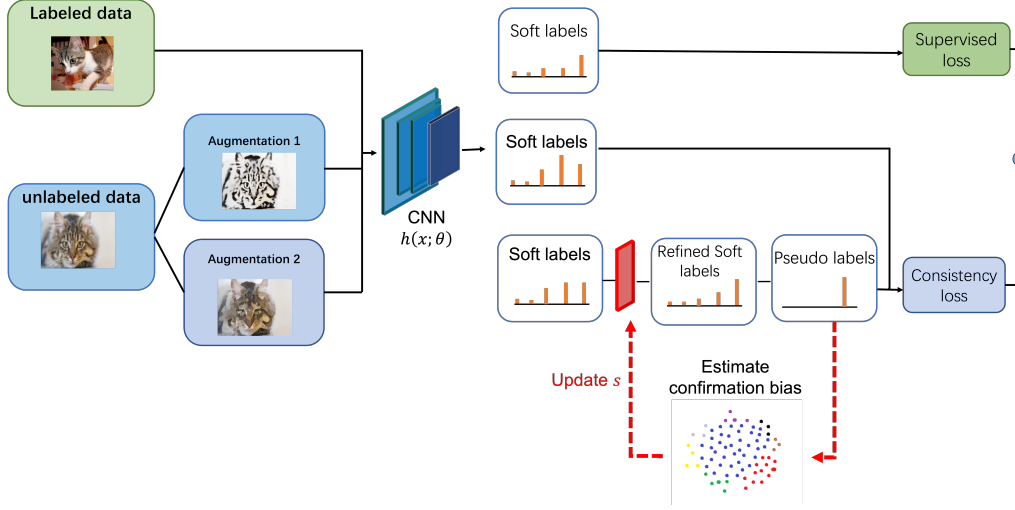


Figure 4.5. Illustration of the SaR framework.

one can make the value of  $s_k$  proportional to the biased degree. In [18], the degree of bias is estimated by counting the number of samples  $\hat{n}_k$  whose predictions fall into class  $k$  and above the threshold, i.e.,

$$\hat{n}_k = \sum_{j=1}^n \mathbf{1}(h(\alpha(x_j^{(u)}); \theta)_k > \tau) \cdot \mathbf{1}(\arg \max(\hat{y}_j^{(u)}) = k), \quad (4.10)$$

where  $\tau$  is pre-defined threshold and  $\mathbf{1}(\cdot)$  is the indicated function. In other words, the class that has more samples with prediction confidence exceeds  $\tau$  is considered to exhibit a lower bias [18]. Hence we can design our Mitigating Vector as follows:

$$s_k = b(\hat{n}_k), k = 1, \dots, C. \quad (4.11)$$

Here  $b(\cdot)$  is a monotone decreasing function. Unlike the FlexMatch [18], we fix the threshold  $\tau$  during the training and focus on refining the soft labels. Besides, blindly matching the distribution of the pseudo labels to the true distribution could bias the classifier [32]. We design a mitigating vector to smooth the distribution of the pseudo labels by balancing the needs of representing the true distribution and emphasizing the minority classes. Specifically, we use the function  $b$ , which is inspired by the effective samples [164] proposed for SL, i.e.,

$$b(\hat{n}_k) = (1 - \beta)/(1 - \beta^{\hat{n}_k}). \quad (4.12)$$

The degree of flattening is determined by the value of  $\beta \in (0, 1)$ . Here  $\beta = 0$  nullifies the impact on the sample counts  $\hat{n}_k$  making all  $s_k$ 's equal to 1. When  $\beta$  goes to 1,  $b(\hat{n}_k)$  returns a

---

**Algorithm 4** The SaR Framework

---

**Data:** Labeled data  $\{(x_i^{(1)}, y_i^{(1)})\}_{i=1}^m$ , unlabeled data  $\{x_j^{(u)}\}_{j=1}^n$ , number of classes  $C$ , learning rate  $\eta$ , max epoch  $T$

Initialize model parameter  $\theta$ , Mitigating Vector  $s$

**while**  $t < T$  **do**

**for**  $1, \dots, K$  **do**

    Sample batches label data  $\{x^{(1)}, y^{(1)}\}$

    Sample batches unlabeled data  $\{x^{(u)}\}$

    Refine the soft labels from unlabeled data with  $s$

    Compute pseudo labels  $p(x^{(u)}; \theta)$  and save them

    Update  $\theta$  by stochastic gradient decent with fixed  $s$

**end for**

  Estimate  $(\hat{n}_1, \dots, \hat{n}_C)$  based on (4.10)

  Update  $s_k = b(\hat{n}_k), k = 1, \dots, C$

**end while**

**Return:**  $\theta$

---

larger value for a smaller  $\hat{n}_k$ , thereby putting more emphasis on the minority classes. We select  $\beta$  as 0.999 and fix it as suggested by [164]. After we refine the soft labels, we will generate a new set of pseudo labels that alleviate the confirmation bias. The entire SaR framework is summarized in Figure 4.5 and algorithm 4.

## 4.3 Experiments

In this section, we evaluate the efficacy of our proposed algorithm on various class-imbalanced scenarios for semi-supervised learning in classification problems, including CIFAR-10, CIFAR-100 [143], and STL-10 [176]. The evaluation criteria on these imbalanced classification problems include *balanced accuracy* (bACC) [167, 156] and *geometric mean scores* (GM) [168, 169]. Each algorithm is run three times to report the mean value and standard deviation under each setting. All experiments use the same encoder (Wide ResNet-28-2 [31]) and batch size (64) under the same number of training iterations (around  $2.5 \times 10^5$ ). More details are in Section A



of the supplementary material.

### 4.3.1 CIFAR10-LT

**Setup.** We first evaluate our approach on a “synthetically long-tailed” variant of CIFAR-10 (CIFAR10-LT) introduced in [32, 164, 170]. In the training set, all images are randomly sampled based on the pre-defined imbalance ratios:  $\gamma_l$  is for labeled data while  $\gamma_u$  is for unlabeled data. As denoted in Section 4.2.1, we set  $m_1 = 1500$  and  $n_1 = 3000$ , then have  $m_k = m_1 \cdot \gamma_l^{\epsilon_k}$  and  $n_k = n_1 \cdot \gamma_u^{\epsilon_k}$ , where  $\epsilon_k = -\frac{k-1}{C-1}$ . After that, we vary  $\gamma_l$  and  $\gamma_u$  to construct multiple CIFAR10-LT to mimic different imbalanced scenarios. The hold-out test set remains untouched and balanced [32]. The results for unbalanced test set are reported in Section 4.3.3.

**Main results under  $\gamma_l = \gamma_u$ .** First, we evaluate our framework assuming that labeled and unlabeled sets are sampled from almost the same distributions by setting  $\gamma$  at 50, 100, 150, respectively [32]. To verify the effectiveness of our framework, we apply it to the recent state-of-the-art SSL algorithms: ReMixMatch [40] and FixMatch [17]. We compare our framework with state-of-the-art imbalanced SSL algorithm, DARP [32]. The results are summarized in Table 4.1. Our proposed framework consistently improves both FixMatch and ReMixMatch on bACC and GM for all evaluated settings. Specifically, SAW improves the bACC of FixMatch up to 6% absolute bACC gain and achieves up to 9.8% absolute GM gain compared to the original FixMatch. Besides, it also improves up to 4.8%/6.5% at absolute bACC/GM gain compared to the original ReMixMatch. And our framework consistently outperforms the recent imbalanced algorithm (DARP) for both bACC and GM in all settings.

**Main results under  $\gamma_l \neq \gamma_u$ .** In real-world scenarios, it is natural that unlabeled data have different distributions, i.e.,  $\gamma_l \neq \gamma_u$ . Table 4.2 summarizes the performance of different algorithms under unknown distributions of unlabeled data. Our proposed framework shows its superiority when unlabeled data have different imbalanced ratio compared to labeled data, especially when unlabeled data are also severely imbalanced (such as  $\gamma_u = 50/150$ ). In these settings, SAW can improve original ReMixMatch up to 5.2% of bACC and 7.7% of GM. Meanwhile, it improves original FixMatch up to 15% of bACC and 40.5% of GM. Even compared to DARP, SAW outperforms up to 4.2% of bACC and 5.9% of GM. Although the performance of SAW on balanced unlabeled data is slightly lower than DARP, it still improves

Table 4.1: Comparison of classification performance on CIFAR10-LT under  $\gamma = \gamma_l = \gamma_u$  (hold-out test set is balanced). The evaluation criterion is bACC/GM. The best results are in bold.

| Algorithm                    | SSL | RB | $\gamma = 50$                                   | $\gamma = 100$                                  | $\gamma = 150$                                  |
|------------------------------|-----|----|-------------------------------------------------|-------------------------------------------------|-------------------------------------------------|
| Wide ResNet-28-2 [31]        | -   | -  | 65.2 $\pm$ 0.05 / 61.1 $\pm$ 0.09               | 58.8 $\pm$ 0.13 / 51.0 $\pm$ 0.11               | 55.6 $\pm$ 0.43 / 44.0 $\pm$ 0.98               |
| Re-sampling [177]            | -   | ✓  | 64.3 $\pm$ 0.48 / 60.6 $\pm$ 0.67               | 55.8 $\pm$ 0.47 / 45.1 $\pm$ 0.30               | 52.2 $\pm$ 0.05 / 38.2 $\pm$ 1.49               |
| LDAM-DRW [161]               | -   | ✓  | 68.9 $\pm$ 0.07 / 67.0 $\pm$ 0.08               | 62.8 $\pm$ 0.17 / 58.9 $\pm$ 0.60               | 57.9 $\pm$ 0.20 / 50.4 $\pm$ 0.30               |
| cRT [172]                    | -   | ✓  | 67.8 $\pm$ 0.13 / 66.3 $\pm$ 0.15               | 63.2 $\pm$ 0.45 / 59.9 $\pm$ 0.40               | 59.3 $\pm$ 0.10 / 54.6 $\pm$ 0.72               |
| ReMixMatch [40]              | ✓   | -  | 81.5 $\pm$ 0.26 / 80.2 $\pm$ 0.32               | 73.8 $\pm$ 0.38 / 69.5 $\pm$ 0.84               | 69.9 $\pm$ 0.47 / 62.5 $\pm$ 0.35               |
| ReMixMatch + cRT [174]       | ✓   | ✓  | 86.8 $\pm$ 0.50 / 86.5 $\pm$ 0.49               | 81.4 $\pm$ 0.41 / 80.7 $\pm$ 0.45               | 78.9 $\pm$ 0.84 / 77.8 $\pm$ 0.94               |
| ReMixMatch + DARP [32]       | ✓   | -  | 82.1 $\pm$ 0.14 / 80.8 $\pm$ 0.09               | 75.8 $\pm$ 0.09 / 72.6 $\pm$ 0.24               | 71.0 $\pm$ 0.27 / 64.5 $\pm$ 0.68               |
| ReMixMatch + DARP + cRT [32] | ✓   | ✓  | 87.3 $\pm$ 0.61 / 87.0 $\pm$ 0.11               | 83.5 $\pm$ 0.07 / 83.1 $\pm$ 0.09               | 79.7 $\pm$ 0.54 / 78.9 $\pm$ 0.49               |
| ReMixMatch + CReST [33]      | ✓   | ✓  | 85.2 $\pm$ 0.19 / 84.9 $\pm$ 0.25               | 76.2 $\pm$ 0.31 / 75.1 $\pm$ 0.28               | 71.4 $\pm$ 0.23 / 67.5 $\pm$ 0.40               |
| ReMixMatch + SaR [42]        | ✓   | -  | 85.1 $\pm$ 0.25 / 84.6 $\pm$ 0.37               | 77.2 $\pm$ 0.71 / 75.5 $\pm$ 0.30               | 72.9 $\pm$ 0.45 / 69.1 $\pm$ 0.23               |
| ReMixMatch + SAW             | ✓   | -  | 86.3 $\pm$ 0.61 / 86.1 $\pm$ 0.64               | 77.0 $\pm$ 0.59 / 76.0 $\pm$ 0.42               | 71.5 $\pm$ 0.30 / 68.9 $\pm$ 0.26               |
| ReMixMatch + SAW + cRT [174] | ✓   | ✓  | <b>87.6</b> $\pm$ 0.21 / <b>87.4</b> $\pm$ 0.26 | <b>85.4</b> $\pm$ 0.32 / <b>83.9</b> $\pm$ 0.21 | <b>79.9</b> $\pm$ 0.15 / <b>79.9</b> $\pm$ 0.12 |
| FixMatch [17]                | ✓   | -  | 79.2 $\pm$ 0.33 / 77.8 $\pm$ 0.36               | 71.5 $\pm$ 0.72 / 66.8 $\pm$ 1.51               | 68.4 $\pm$ 0.15 / 59.9 $\pm$ 0.43               |
| FixMatch + DARP [32]         | ✓   | -  | 81.8 $\pm$ 0.24 / 80.9 $\pm$ 0.28               | 75.5 $\pm$ 0.05 / 73.0 $\pm$ 0.09               | 70.4 $\pm$ 0.25 / 64.9 $\pm$ 0.17               |
| FixMatch + CReST [33]        | ✓   | ✓  | 83.0 $\pm$ 0.39 / 81.5 $\pm$ 0.17               | 75.7 $\pm$ 0.38 / 72.7 $\pm$ 0.85               | 70.8 $\pm$ 0.25 / 64.5 $\pm$ 0.31               |
| FixMatch + CReST + & LA [33] | ✓   | ✓  | 85.6 $\pm$ 0.36 / 81.9 $\pm$ 0.45               | <b>81.2</b> $\pm$ 0.70 / 74.5 $\pm$ 0.99        | 71.9 $\pm$ 2.24 / 64.4 $\pm$ 1.75               |
| FixMatch + SaR [42]          | ✓   | -  | 83.9 $\pm$ 0.09 / 82.9 $\pm$ 0.15               | 77.6 $\pm$ 0.42 / 75.9 $\pm$ 0.76               | 71.5 $\pm$ 0.23 / 66.9 $\pm$ 0.25               |
| FixMatch + SAW               | ✓   | -  | 84.0 $\pm$ 0.10 / 83.6 $\pm$ 0.12               | 77.5 $\pm$ 0.63 / 76.3 $\pm$ 0.80               | 71.6 $\pm$ 0.35 / 69.7 $\pm$ 0.46               |
| FixMatch + SAW + & LA [171]  | ✓   | ✓  | <b>86.2</b> $\pm$ 0.15 / <b>83.9</b> $\pm$ 0.35 | 80.7 $\pm$ 0.15 / <b>77.5</b> $\pm$ 0.21        | <b>73.7</b> $\pm$ 0.06 / <b>71.2</b> $\pm$ 0.17 |

Table 4.2: Comparison of classification performance on CIFAR10-LT under  $\gamma_l = 100, \gamma_u \neq \gamma_l$  (hold-out test set is balanced). The evaluation criterion is bACC/GM. The best results are in bold.

| Algorithm               | CIFAR10-LT ( $\gamma_l = 100$ )                 |                                                 |                                                 |
|-------------------------|-------------------------------------------------|-------------------------------------------------|-------------------------------------------------|
|                         | $\gamma_u = 1$                                  | $\gamma_u = 50$                                 | $\gamma_u = 150$                                |
| ReMixMatch [40]         | 48.3 $\pm$ 0.14 / 19.5 $\pm$ 0.85               | 75.1 $\pm$ 0.43 / 71.9 $\pm$ 0.77               | 72.5 $\pm$ 0.10 / 68.2 $\pm$ 0.32               |
| ReMixMatch*             | 85.0 $\pm$ 1.35 / 84.3 $\pm$ 1.55               | 77.0 $\pm$ 0.12 / 74.7 $\pm$ 0.04               | 72.8 $\pm$ 0.10 / 68.8 $\pm$ 0.21               |
| ReMixMatch* + DARP [32] | <b>89.7</b> $\pm$ 0.15 / <b>89.4</b> $\pm$ 0.17 | 77.4 $\pm$ 0.22 / 75.0 $\pm$ 0.25               | 73.2 $\pm$ 0.11 / 69.2 $\pm$ 0.31               |
| ReMixMatch* + SAW       | 88.3 $\pm$ 0.15 / 88.9 $\pm$ 0.10               | <b>80.3</b> $\pm$ 0.36 / <b>79.6</b> $\pm$ 0.40 | <b>74.0</b> $\pm$ 0.94 / <b>72.4</b> $\pm$ 0.94 |
| FixMatch [17]           | 68.9 $\pm$ 1.95 / 42.8 $\pm$ 8.11               | 73.9 $\pm$ 0.25 / 70.5 $\pm$ 0.52               | 69.6 $\pm$ 0.60 / 62.6 $\pm$ 1.11               |
| FixMatch + DARP [32]    | <b>85.4</b> $\pm$ 0.55 / <b>85.0</b> $\pm$ 0.65 | 77.3 $\pm$ 0.17 / 75.5 $\pm$ 0.21               | 72.9 $\pm$ 0.24 / 69.5 $\pm$ 0.18               |
| FixMatch + SAW          | 83.9 $\pm$ 0.44 / 83.3 $\pm$ 0.47               | <b>81.5</b> $\pm$ 2.25 / <b>80.9</b> $\pm$ 2.30 | <b>76.8</b> $\pm$ 0.31 / <b>75.4</b> $\pm$ 0.37 |

original FixMatch and ReMixMatch significantly. Unlike DARP [32], SAW does not require any prior information from unlabeled data, and our experiments show that SAW can be applied in a broader range of imbalanced scenarios.

### 4.3.2 CIFAR100-LT and STL-10

**Setup.** Besides CIFAR10-LT, we also evaluate our framework on CIFAR-100 [143] and STL-10 [176]. For CIFAR-100, we follow the process outlined in Section 4.3.1 to set  $\gamma$  and prepare a long-tailed training set as CIFAR100-LT, where we set  $m_1 = 150$  and  $n_1 = 300$ . For STL-10, we set  $m_1$  as 450 and construct a similar “synthetically long-tailed” version of labeled data; all unlabeled data are used since their class distribution is unknown. Hence the imbalanced ratio of labeled data can be different from that of unlabeled data ( $\gamma_l \neq \gamma_u$ ).

**Main results.** The results on CIFAR10-LT and STL-10 are summarized in Table 4.3. Note that the gain of SAW is significant under  $\gamma_l = 20$  in STL-10: 19.1%/30.0% of improvements at bACC/GM for ReMixMatch, 18.5%/16.4% of improvements at bACC/GM for FixMatch. When we compare it with DARP, it still outperforms DARP up to 8.3%/10.9% on ReMixMatch. This is

Table 4.3: Comparison of classification performance on CIFAR100-LT under  $\gamma = \gamma_l = \gamma_u$ , and STL-10 where the distribution of unlabeled data is unknown. Hold-out test set is balanced. The evaluation criterion is bACC/GM. The best results are in bold.

| Algorithm          | CIFAR100-LT ( $\gamma = \gamma_l = \gamma_u$ )                    |                                                                   | STL-10                                                            |                                                                   |
|--------------------|-------------------------------------------------------------------|-------------------------------------------------------------------|-------------------------------------------------------------------|-------------------------------------------------------------------|
|                    | $\gamma = 10$                                                     | $\gamma = 20$                                                     | $\gamma_l = 10$                                                   | $\gamma_l = 20$                                                   |
| ReMixMatch [40]    | 59.2 $\pm$ 0.03 / 52.1 $\pm$ 0.13                                 | 53.5 $\pm$ 0.03 / 42.3 $\pm$ 0.13                                 | 67.8 $\pm$ 0.45 / 61.1 $\pm$ 0.92                                 | 60.1 $\pm$ 1.18 / 44.9 $\pm$ 1.52                                 |
| ReMixMatch* + DARP | 59.8 $\pm$ 0.20 / 52.9 $\pm$ 0.41                                 | 54.4 $\pm$ 0.07 / 44.2 $\pm$ 0.07                                 | 79.4 $\pm$ 0.07 / 78.2 $\pm$ 0.10                                 | 70.9 $\pm$ 0.44 / 67.0 $\pm$ 1.62                                 |
| ReMixMatch* + SAW  | <b>61.8<math>\pm</math>0.06</b> / <b>56.9<math>\pm</math>0.40</b> | <b>55.3<math>\pm</math>0.26</b> / <b>46.3<math>\pm</math>0.65</b> | <b>82.0<math>\pm</math>0.55</b> / <b>81.0<math>\pm</math>0.64</b> | <b>79.2<math>\pm</math>0.44</b> / <b>77.9<math>\pm</math>0.52</b> |
| FixMatch [17]      | 60.1 $\pm$ 0.05 / 54.4 $\pm$ 0.11                                 | 54.0 $\pm$ 0.04 / 44.4 $\pm$ 0.17                                 | 72.9 $\pm$ 0.09 / 69.6 $\pm$ 0.01                                 | 63.4 $\pm$ 0.21 / 52.6 $\pm$ 0.09                                 |
| FixMatch + DARP    | 61.1 $\pm$ 0.23 / 56.4 $\pm$ 0.28                                 | 54.9 $\pm$ 0.05 / 46.4 $\pm$ 0.41                                 | 77.8 $\pm$ 0.33 / 76.5 $\pm$ 0.40                                 | 69.9 $\pm$ 1.77 / 65.4 $\pm$ 3.07                                 |
| FixMatch + SAW     | <b>62.1<math>\pm</math>0.25</b> / <b>58.0<math>\pm</math>0.20</b> | <b>55.7<math>\pm</math>0.10</b> / <b>49.4<math>\pm</math>0.40</b> | <b>78.3<math>\pm</math>0.25</b> / <b>77.0<math>\pm</math>0.19</b> | <b>71.9<math>\pm</math>0.81</b> / <b>69.0<math>\pm</math>0.81</b> |

due to the vast mismatch between the distribution of labeled data and that of unlabeled data. It is noticeable that SAW achieves significant improvements on STL-10 that has a tailored unlabeled set for SSL, which indicates there is no prior information on the unlabeled data of STL-10. Therefore, the results here prove the superiority of SAW for both matching and non-matching distributions between the labeled and unlabeled data.

### 4.3.3 Empirical analysis

**Stress-test.** The previous sections report the results on a balanced test set. In reality, if the training set is severely imbalanced, the hold-out test set can also be imbalanced, and even follow a totally different distribution compared to the training set. To further evaluate the robustness of SAW, we add a stress-test: re-sample the original balanced hold-out test set based on the flipped distribution of the long-tail unlabeled data. Hence the stress-test set has totally reversed distributions compared to unlabeled training data. Table 4.4 summarizes the results on the situations where labeled and unlabeled data are sampled from the same distribution. Our SAW greatly improves both original SSL algorithms and outperforms DARP [32]. SAW improves ReMixMatch up to 30% in terms of bACC and 2.6% at GM. Similarly, SAW also improves original FixMatch up to 19% at bACC and 7.1% at GM. We also find that DARP may degrade

Table 4.4: Comparison of classification performance on CIFAR10-LT under  $\gamma = \gamma_l = \gamma_u$  (hold-out test set is of reversed distribution). The evaluation criterion is bACC/GM. The best results are in bold.

| Algorithm              | CIFAR-10                                        |                                                 |                                                 |
|------------------------|-------------------------------------------------|-------------------------------------------------|-------------------------------------------------|
|                        | $\gamma = 50$                                   | $\gamma = 100$                                  | $\gamma = 150$                                  |
| ReMixMatch [40]        | 71.0 $\pm$ 0.55 / 83.5 $\pm$ 0.29               | 54.7 $\pm$ 0.51 / 74.4 $\pm$ 0.47               | 41.5 $\pm$ 1.69 / 66.4 $\pm$ 1.22               |
| ReMixMatch + DARP [32] | 66.9 $\pm$ 0.75 / 80.5 $\pm$ 0.46               | 49.7 $\pm$ 1.55 / 70.5 $\pm$ 0.90               | 35.8 $\pm$ 1.81 / 60.9 $\pm$ 2.42               |
| ReMixMatch + SAW       | <b>86.3</b> $\pm$ 0.61 / <b>86.1</b> $\pm$ 0.64 | <b>77.0</b> $\pm$ 0.59 / <b>76.0</b> $\pm$ 0.42 | <b>71.5</b> $\pm$ 0.30 / <b>68.9</b> $\pm$ 0.26 |
| FixMatch [17]          | 70.5 $\pm$ 0.26 / 82.2 $\pm$ 0.31               | 51.0 $\pm$ 1.65 / 71.5 $\pm$ 1.24               | 38.5 $\pm$ 1.15 / 63.4 $\pm$ 0.31               |
| FixMatch + DARP [32]   | 72.2 $\pm$ 0.62 / 82.8 $\pm$ 0.17               | 57.6 $\pm$ 0.36 / 74.8 $\pm$ 0.48               | 46.5 $\pm$ 1.26 / 68.1 $\pm$ 0.10               |
| FixMatch + SAW         | <b>78.7</b> $\pm$ 0.77 / <b>84.2</b> $\pm$ 0.36 | <b>64.3</b> $\pm$ 1.96 / <b>76.4</b> $\pm$ 0.88 | <b>57.5</b> $\pm$ 2.83 / <b>70.5</b> $\pm$ 1.50 |

the performance of original ReMixMatch while SAW consistently outperforms both DARP and baselines.

We also conduct stress-test experiments under the settings of  $\gamma_l \neq \gamma_u$  on the test set of reversed distributions. The results are summarized in Table 4.5. SAW achieves significant improvements in all settings where the distribution of unlabeled data is different from that of labeled data. Specifically, SAW improves original FixMatch by up to 15.4% of bACC and 35.1% of GM; SAW also improves original ReMixMatch by up to 14.0% of bACC and 35.4% of GM. SAW also shows its superiority compared to DARP, especially considering SAW does not require any prior information from the unlabeled data.

**Per-class performance.** To show the source of our absolute performance gain, we present per-class F1 score [178] on the balanced test set of CIFAR10-LT with  $\gamma_l = \gamma_u = 100$  as shown in Figure 4.6(a). Note that SAW improves the F1 score for almost every class, especially for minority classes (index as 8,9,10). Specifically, SAW provides gain on the precision of majority classes and the recall of minority classes without much penalty on other metrics. This explains again why SAW achieves significant gain in Table 4.4 where the distribution of hold-out test set

Table 4.5: Comparison of classification performance on CIFAR10-LT under  $\gamma_l = 100, \gamma_u \neq \gamma_l$  (hold-out test set is of reversed distributions). The evaluation criterion is bACC/GM. The best results are in bold.

| CIFAR-10 ( $\gamma_l = 100$ ) |                                                                   |                                                                   |                                                                   |
|-------------------------------|-------------------------------------------------------------------|-------------------------------------------------------------------|-------------------------------------------------------------------|
| Algorithm                     | $\gamma_u = 1$                                                    | $\gamma_u = 50$                                                   | $\gamma_u = 150$                                                  |
| ReMixMatch [40]               | 70.0 $\pm$ 1.39 / 48.3 $\pm$ 0.53                                 | 61.4 $\pm$ 0.21 / 77.3 $\pm$ 0.41                                 | 48.6 $\pm$ 1.62 / 71.3 $\pm$ 0.67                                 |
| ReMixMatch* + DARP [32]       | 67.3 $\pm$ 1.63 / 49.5 $\pm$ 2.01                                 | 54.5 $\pm$ 1.30 / 72.6 $\pm$ 0.58                                 | 44.6 $\pm$ 1.69 / 67.7 $\pm$ 1.16                                 |
| ReMixMatch* + SAW             | <b>84.0<math>\pm</math>0.05</b> / <b>83.7<math>\pm</math>0.42</b> | <b>72.7<math>\pm</math>0.56</b> / <b>79.6<math>\pm</math>1.11</b> | <b>59.8<math>\pm</math>1.03</b> / <b>73.6<math>\pm</math>1.19</b> |
| FixMatch [17]                 | 70.0 $\pm$ 1.39 / 48.3 $\pm$ 0.53                                 | 61.0 $\pm$ 1.61 / 76.4 $\pm$ 0.52                                 | 48.0 $\pm$ 2.26 / 70.4 $\pm$ 0.88                                 |
| FixMatch + DARP [32]          | 68.1 $\pm$ 0.37 / 48.5 $\pm$ 1.46                                 | 59.5 $\pm$ 1.41 / 75.2 $\pm$ 1.08                                 | 55.1 $\pm$ 1.31 / 73.8 $\pm$ 0.43                                 |
| FixMatch + SAW                | <b>84.0<math>\pm</math>0.39</b> / <b>83.4<math>\pm</math>0.42</b> | <b>76.4<math>\pm</math>6.71</b> / <b>81.5<math>\pm</math>1.97</b> | <b>63.3<math>\pm</math>0.37</b> / <b>75.4<math>\pm</math>1.23</b> |

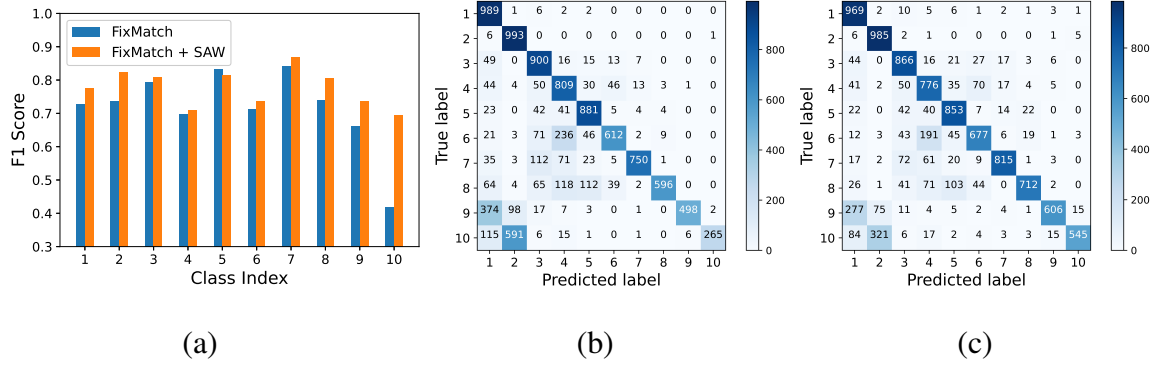


Figure 4.6. Per-class results on the balanced test set of CIFAR-10 under the imbalanced ratio  $\gamma = 100$ . (a) F1 score for each class. (b) Confusion matrix of original FixMatch. (c) Confusion matrix of FixMatch with SAW.

is flipped based on unlabeled data' distribution (the minority class in the training set becomes the majority class in the test set). Figure 4.6 also shows the comparison of the confusion matrix of original FixMatch (Figure 4.6(b)) and SAW (Figure 4.6(c)) on the hold-out test set. Similar to the estimated distributions shown in Figure 4.1(b) and (c), we find that SAW provides robust re-weighting on both the minority and majority classes due to our smoothed weighting scheme, which is easily transferred to better generalization on a test set of totally different distributions.

We also compare the per-class performance during the training process between original FixMatch [17] and SAW. As shown in Figure 4.7, we plot the confusion matrix after training 100, 200, and 300 epochs, respectively. SAW improves the performance of minority classes quickly, even 100 epochs make a difference compared to the original FixMatch.

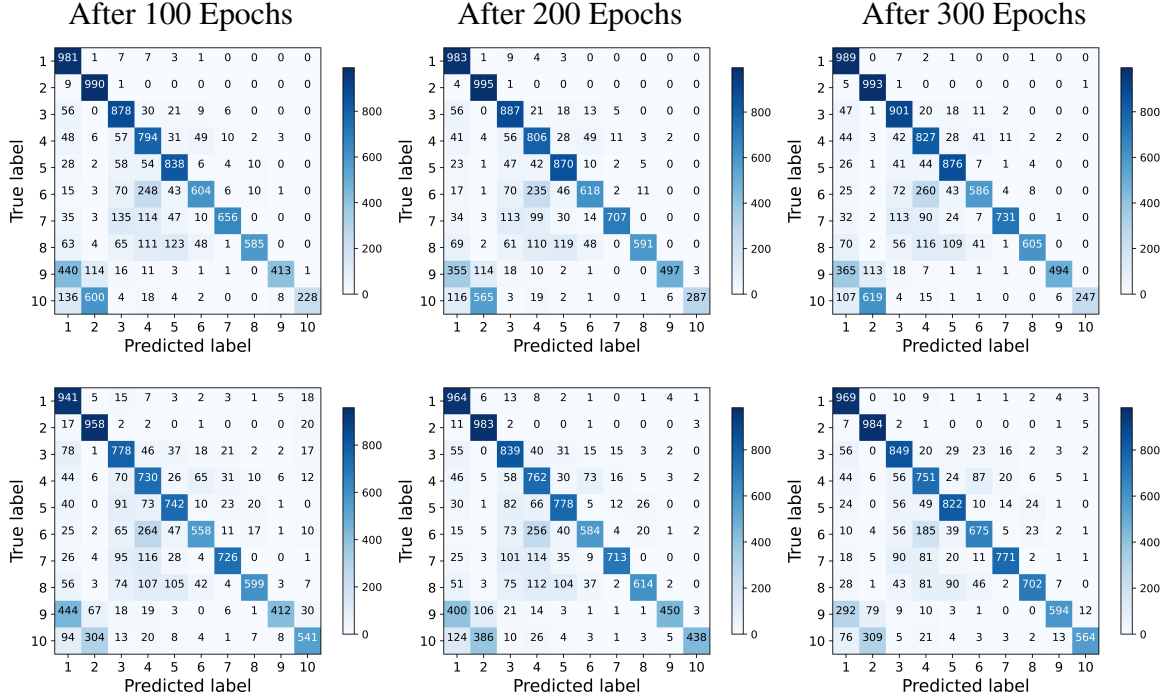


Figure 4.7. The confusion matrix comparison between original FixMatch [17] and FixMatch + SAW during the training progress. The top row is from original FixMatch while the bottom row denotes FixMatch + SAW.

**Effect of the parameter  $\beta$ .** As discussed in Section 4.2, we heuristically select  $\beta =$  Table 4.6: Comparison of classification performance on CIFAR10-LT under  $\gamma = 100$  (hold-out test set is of balanced) with a fixed  $\beta = 0.99, 0.999, 0.9999$ . The evaluation criterion is bACC/GM. The best results are in bold.

| Algorithm      | $\beta = 0.99$                    | $\beta = 0.999$                                 | $\beta = 0.9999$                  |
|----------------|-----------------------------------|-------------------------------------------------|-----------------------------------|
| FixMatch + SW  | 75.0 $\pm$ 0.67 / 73.0 $\pm$ 0.58 | <b>77.5</b> $\pm$ 0.98 / <b>76.3</b> $\pm$ 1.21 | 72.4 $\pm$ 1.69 / 70.7 $\pm$ 2.16 |
| FixMatch + SAW | 77.3 $\pm$ 0.67 / 75.4 $\pm$ 0.88 | <b>77.4</b> $\pm$ 0.30 / <b>76.2</b> $\pm$ 0.40 | 73.7 $\pm$ 0.63 / 72.3 $\pm$ 0.93 |

$(N - 1)/N$ , where  $N$  is updated after each epoch via  $N = \sum_{k=1}^C (m_k + \hat{n}_k)/C$ . Here a sensitivity analysis on the parameter  $\beta$  is provided. Similarly, we consider CIFAR10-LT under  $\gamma = 100$ , and test both SW and SAW with a fixed  $\beta \in \{0.99, 0.999, 0.9999\}$  in the training process. Table 4.6 shows that the trend is similar to the weighting function of  $\nu$ , indicating the effectiveness of smoothed weighting in SSL. Moreover, SAW is less sensitive to the value of  $\beta$ . Therefore, there is no need to specifically tune the parameter  $\beta$  for each setting. The heuristic selection in SAW already brings the optimal result among these trials.

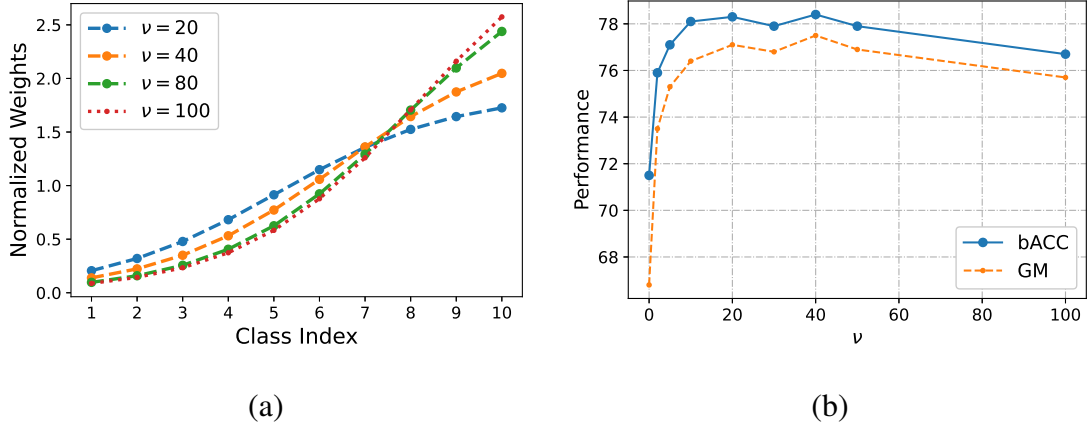


Figure 4.8. Alternative for weighting function in SAW. (a) The normalized weights with different values of  $\nu$ . (b) bACC and GM of FixMatch-SAW in the alternative weighting function.

**Alternative for smoothed weighting function and degree of smoothing.** While it is key that the weights in SSL should be smoothed, it is feasible that other smoothed weighting schemes may also work. Here we show an alternate instantiation of weighting function to gain more insights. Instead of choosing weights based on the effective samples (4.4), we define another smoothed weighting function:

$$w_k \propto \frac{n}{n + \nu \cdot \hat{n}_k}, \text{ for } k = 1, \dots, C. \quad (4.13)$$

where  $\nu$  determines the degree of smoothing. Note that  $\nu = 0$  refers uniform weights while  $\nu \rightarrow \infty$  corresponds to the weight by strictly inverse class frequency. Compared to (4.4), this function provides wider range to re-tune the smoothing degree. We set the imbalance ratio of CIFAR10-LT as  $\gamma_l = \gamma_u = 100$  and test  $\nu$  between 0 to 100. Figure 4.8 shows the performance with different value of  $\nu$ . to decrease the ratio between the weight of minority class and that of



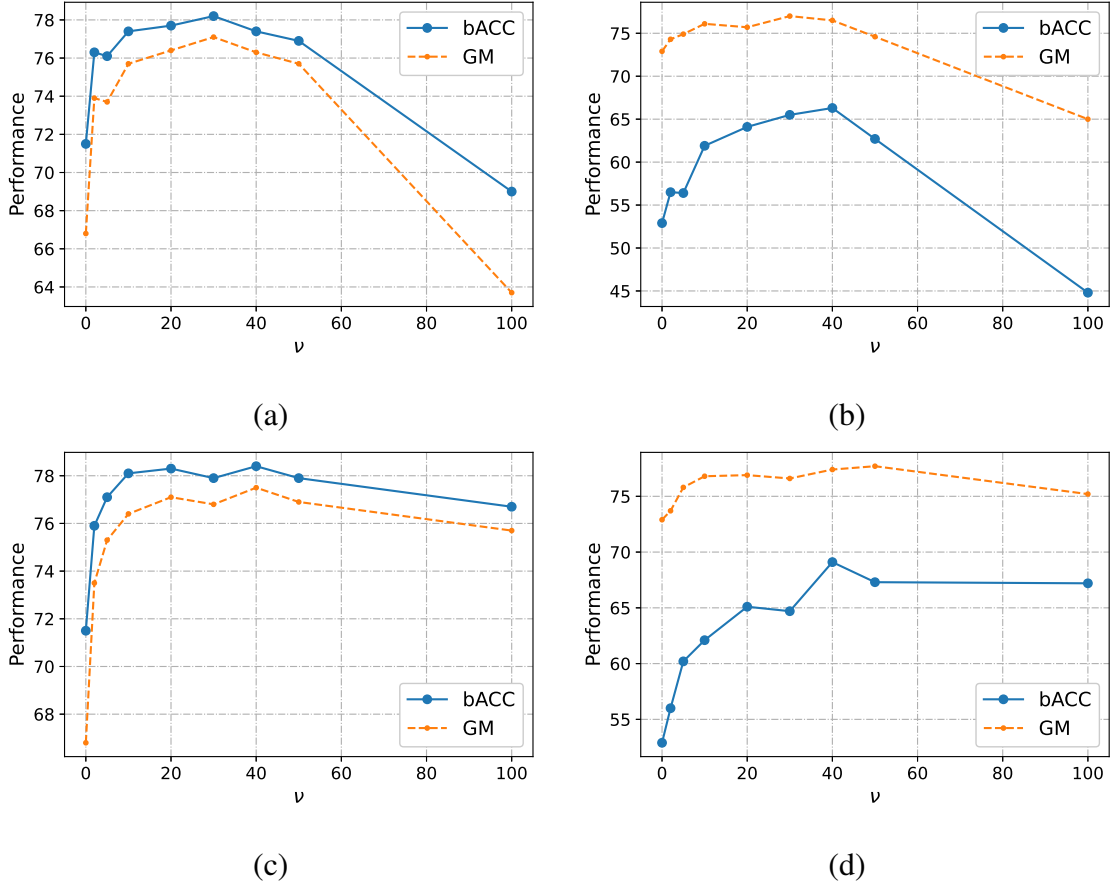


Figure 4.9. Performance comparison with the value of  $\nu$  in the alternative smoothed weighting function. (a) without normalized weights tested on a balanced test set, (b) without normalized weights tested on a reversed test set, (c) with normalized weights tested on a balanced test set, (d) with normalized weights tested on a reversed test set.

majority class from 100 : 1 (original frequency ratio) to 50 : 1 until 1 : 1 (uniform weights, same as the original FixMatch). Based on our observations, the degree of smoothing is not sensitive to the performance, while  $\nu \in [10, 40]$  could achieve optimal results, which are compatible to the results in the weighting function (4.4). This proves the advantages of smoothing weights in imbalanced SSL.

To comprehensively analyze where the model achieves the optimal performance, we test it on two test sets of different distributions: balanced distribution and reversed distribution. We conclude that  $\nu \in [10, 40]$  could achieve the optimal results. Another observation we have is the effect of normalization on smoothed weights: without normalization, the performance could drastically degrade with a very large  $\nu$ . This is because the loss function of SSL is made up of

both labeled loss and unlabeled loss, unnormalized weights could re-balance the importance between labeled loss and unlabeled loss. To keep these two loss items balanced as the same setting of SSL models, we adopt the normalized weights in our proposed framework.

**Effect of adaptive updating.** The distribution of unlabeled data are usually unavailable, which poses a challenge to define the effective sample and weight for each imbalanced class. SAW alternatively estimate class distribution directly from unlabeled data and update class weights. To investigate the effect of adaptive updating, we conduct a sanity test: provide the true class distribution to the model by replacing the estimated distribution with the true distribution. Table 4.7: Comparison of classification performance on CIFAR10-LT under  $\gamma = \gamma_l = \gamma_u$  (hold-out test set is balanced). The evaluation criterion is bACC. The best results are in bold.

| Algorithm              | CIFAR10-LT             |                        |                        |
|------------------------|------------------------|------------------------|------------------------|
|                        | $\gamma = 50$          | $\gamma = 100$         | $\gamma = 150$         |
| FixMatch [17]          | 79.2 $\pm$ 0.33        | 71.5 $\pm$ 0.72        | 68.4 $\pm$ 0.15        |
| FixMatch + DARP [32]   | 81.8 $\pm$ 0.24        | 75.5 $\pm$ 0.05        | 70.4 $\pm$ 0.25        |
| FixMatch + CReST [33]  | 83.0 $\pm$ 0.39        | 75.7 $\pm$ 0.38        | 70.8 $\pm$ 0.25        |
| FixMatch + CReST+ [33] | 83.9 $\pm$ 0.14        | 77.4 $\pm$ 0.36        | <b>72.8</b> $\pm$ 0.58 |
| FixMatch + SW          | 82.7 $\pm$ 0.58        | 76.2 $\pm$ 0.06        | 70.8 $\pm$ 0.26        |
| FixMatch + SAW         | <b>84.0</b> $\pm$ 0.10 | <b>77.5</b> $\pm$ 0.63 | 71.6 $\pm$ 0.35        |

Here we test the CIFAR10-LT where the labeled and unlabeled data share the same distribution. The true distribution of unlabeled data can be directly estimated from labeled data. To distinguish from SAW, we call this smoothed weighting without the adaptive scheme as SW. The results are summarized in Table 4.7. With the true distribution for estimating  $\beta$ , our SW improves FixMatch up to 4.7% on bACC when  $\gamma = 100$ . With the adaptive weighting scheme, our algorithm SAW further improves the performance, which proves the advantage of the adaptive weighting. SAW even achieves better performance compared to CReST+ [33] considering they have a strong pre-assumption that unlabeled data are sample from the same distributions as labeled data while SAW does not require any prior information from unlabeled data. Our adaptive scheme shows almost equivalent performance, and even improves the non-adaptive

scheme up to 1.3% (see in the supplementary material) of absolute performance gain, which means our estimated distribution and adaptive updating is helpful for the training process. This proves the superiority of SAW in real-world scenarios where the distributions of unlabeled data are unknown a priori.

**Effect of weights on supervised loss.** The motivation on adding the weight in consistency loss is to alleviate the bias in pseudo labels during the class-imbalanced training. We have studied the function of weighting in both supervised loss and consistency loss. We conclude that applying smoothed weights to both loss items can result in the best performance based on our experiments. Table 4.8 summarizes the results: the weights on consistency loss are important, which is expected considering the fact that the majority samples are unlabeled in SSL settings. We also find that applying weights to the supervised loss can further improve the performance. Thus we decide to apply smoothed weights to both loss items, which is SAW.

Table 4.8: Comparison of classification performance on CIFAR10-LT under  $\gamma = \gamma_l = \gamma_u$  (hold-out test set is balanced). The evaluation criterion is bACC. SL\_W denotes only applying smoothed weights to the supervised loss on labeled data, CL\_W denotes only applying smoothed weights to the consistency loss.

| Algorithm       | CIFAR10-LT      |                 |                 |
|-----------------|-----------------|-----------------|-----------------|
|                 | $\gamma = 50$   | $\gamma = 100$  | $\gamma = 150$  |
| FixMatch [17]   | 79.2 $\pm$ 0.33 | 71.5 $\pm$ 0.72 | 68.4 $\pm$ 0.15 |
| FixMatch + SL_W | 79.9 $\pm$ 0.05 | 72.2 $\pm$ 0.89 | 68.9 $\pm$ 0.33 |
| FixMatch + CL_W | 82.5 $\pm$ 0.30 | 76.3 $\pm$ 0.21 | 70.4 $\pm$ 0.19 |
| FixMatch + SAW  | 84.0 $\pm$ 0.10 | 77.5 $\pm$ 0.63 | 71.6 $\pm$ 0.35 |

## 4.4 Discussion

In this chapter, we have presented the effectiveness of smoothed adaptive weighting scheme (SAW) on imbalanced semi-supervised learning via various imbalanced scenarios using several public datasets. Our main idea is to progressively adapt the weights for the consistency loss based on the class distribution of pseudo labels, which can overcome the challenge of unknown distribution in unlabeled data. We have demonstrated that the smoothing weights are important for imbalanced SSL because they will force the model to focus more on minority classes by introducing moderate penalty. When combined with two leading SSL methods FixMatch and ReMixMatch, SAW shows consistent improvements over the baselines and those augmented with DARP in almost all experiment settings. The two main advantages of SAW are: first, it does not require any knowledge of the class distribution of the unlabeled data and work well even in the case when the labeled data, unlabeled data and testing data all assume vastly different distributions; second, it can be easily integrated with any SSL algorithms that use consistency loss.

# Chapter 5

## Joint Semi-supervised and Active Learning Framework

We have proved the effectiveness of semi-supervised learning in Chapter 3. However, the selection criterion for labeled data in pathology images remains unclear. There are no well-established SSL guidelines on how to determine the amount of labeled data that is needed for the specific task.

Active learning (AL), on the other hand, aims at maximizing a model’s performance with minimal labeling efforts [43]. One challenge of AL is the *cold start problem*: when the selection of starting set is poor or the size is not sufficiently large, the models trained in subsequent cycles can be high-biased, resulting in poor selection again in the following cycles [179, 44]. Another challenge is that the annotation acquisition during the training process may require substantial rounds of labeling, while at the same time, the annotators cannot guarantee unlimited availability during the whole process. This would notably extend the time needed for the entire project. To reduce the number of rounds needed, batch-mode AL methods are proposed [180, 181, 182, 183, 184, 185] to select a batch of data to be labeled at once. However, these methods still require many rounds of labeling and have not been evaluated on histology images.

Therefore, to minimize the labeling efforts, a natural direction is to combine AL and SSL as both of them aim to improve a model’s performance on limited labeled data. More importantly, SSL and AL are complementary in function: SSL has the potential to leverage the unlabeled

data to relieve the *cold start problem* of AL; on the other hand, AL can select the most optimal data to label for SSL. In this paper, we propose a SSL framework with AL-driven region-based labeled data selection [45]. We evaluate our framework on an Amyloid- $\beta$  stained neuropathology images at gigapixel level to separate grey matter (GM) and white matter (WM), which is an important task for analyzing density and distribution of Amyloid- $\beta$  plaques in the brain. Although supervised learning methods [15, 26] have already shown promising results, their performances face a severe degradation when labeled data is limited (as shown in Table 5.2).

The contributions of this section can be summarized as follows:

- We combine SSL and AL to construct a label-efficient learning framework with well-defined selection of regions for annotation to construct a small but effectively labeled dataset.
- To tackle the “cold-start” issue of AL, we design a region-based selection criterion to expand the diversity and volume of the labeled set effectively.
- We evaluate the proposed framework on a pathology segmentation task and it outperforms the state-of-the-art SSL (FixMatch [17]) by more than 10% of IoU score and DICE coefficient [46].

## 5.1 Related work

**Semi-supervised learning.** Semi-supervised learning (SSL) leverages unlabeled data to improve the performance of supervised learning when the amount of labeled data is limited [186]. Classical SSL approaches generate pseudo-labels for each unlabeled sample based on model’s prediction, and use them to train the model via pseudo-labeling [147, 141], consistency regularization [151, 140], and the combination of them [30, 40, 187, 17]. Recent state-of-the-art SSL algorithms such as FixMatch [17] apply regularization consistency on different augmented views of the same unlabeled data, while generating pseudo labels accordingly during the training process. Although these SSL methods achieve promising results, their applicability on histology images is still under study [21]. There are two challenges for these SSL approaches: 1) pseudo labels, generated based on a poorly estimated class distribution of limited labeled data, increases

the difficulty in updating the true distribution of unlabeled data. [34]; 2) another natural question is how to determine the most valuable samples and the size of the samples to be labeled and added to the labeled set.

**Active learning.** Active learning (AL) aims to maximize the performance gain of learning by selecting the fewest possible samples and sending them to annotators for labeling [188, 189]. Recent AL approaches include uncertainty-based, diversity-based, and model performance change-based. The uncertainty-based AL approaches use *max entropy* and *max margin* [190] as the selection criteria due to simplicity. Earlier methods [191, 192, 193, 194, 195, 196, 197] worked on the setting that only one sample is selected for labeling in each cycle of AL. Batch-mode AL methods [180, 181, 182, 183, 184, 185] were proposed later to select a batch of samples at once. However, all of these methods rely on a sufficiently large and representative labeled set as the starting set, otherwise the *cold-start* problem would appear and hurt the performance by biasing the model.

**Semi-supervised active learning.** Although both SSL and AL share a similar goal for minimizing labeling efforts, only a few works have considered combining them. In [198], they combine SSL and AL for speech understanding to reduce significant errors with limited speech data. Rhee *et al.* built a semi-supervised AL system in the pedestrian detection task [199]. Sener *et al.* redefined AL as core-set selection and considered SSL during AL cycles [181]. However, all of these treated SSL and AL independently without considering their mutual impact on each other [200]. In addition, none of them have been evaluated on the histology images, hence their applicability is still questionable.

## 5.2 Methodology

### 5.2.1 Problem setup

We first set up the GM and WM segmentation problem on gigapixel histopathology images. A recent work [201] revealed that per-pixel classification is not all one needs to tackle a semantic segmentation problem. Lai *et al.* [15] transforms this problem into a patch-based classification problem, where one WSI is tiled into a group of patches (more details can be found in Chapter 2). Here, we follow their approach and apply semi-supervised learning to classify each patch into

three categories, i.e., WM, GM, and background. The training data  $\mathcal{D}$  contains a labeled set  $\mathcal{X} = \{(x_i, y_i)\}$ , and an unlabeled dataset  $\mathcal{U} = \{x_j\}$  where  $x_i \in \mathbb{R}^{d \times d}$  is the  $i$ -th patch and  $y_i \in \{0, 1\}^3$  is the corresponding one-hot label. Our goal is to train a classifier  $h(x, \theta) : \mathbb{R}^{d \times d} \rightarrow [0, 1]^3$ , where  $\theta$  is the CNN parameter and the  $k$ -th component of  $h(x, \theta)$  is the predictive probability of the  $k$ -th class for an input  $x$ .  $\theta$  is trained by minimizing an objective loss function:

$$\min_{\theta \in \Theta} L(\mathcal{X}, \theta) + \Omega(\mathcal{D}, \theta), \quad (5.1)$$

where  $L(\mathcal{X}, \theta) := \sum_{(x,y) \in \mathcal{X}} l(x, y, \theta)$  and  $\Omega(\mathcal{D}, \theta) := \sum_{x \in \mathcal{D}} \omega(x, \theta)$ . We denote the per-sample supervised loss and regularization as  $l$  and  $\omega$ , respectively. We mainly adopt FixMatch [17], which generates artificial labels using both pseudo-labeling and consistency regularization. Specifically, the pseudo label is generated based on a weakly-augmented unlabeled image (weak), which will be the target to be compared with the output of the model on a strongly-augmented version of the same unlabeled image (strong). One example of weak and strong augmentation operations is shown in Figure 5.1: weak augmentations include random flip and rotations while strong augmentation include RandAugment [202] and CTAugment [40].

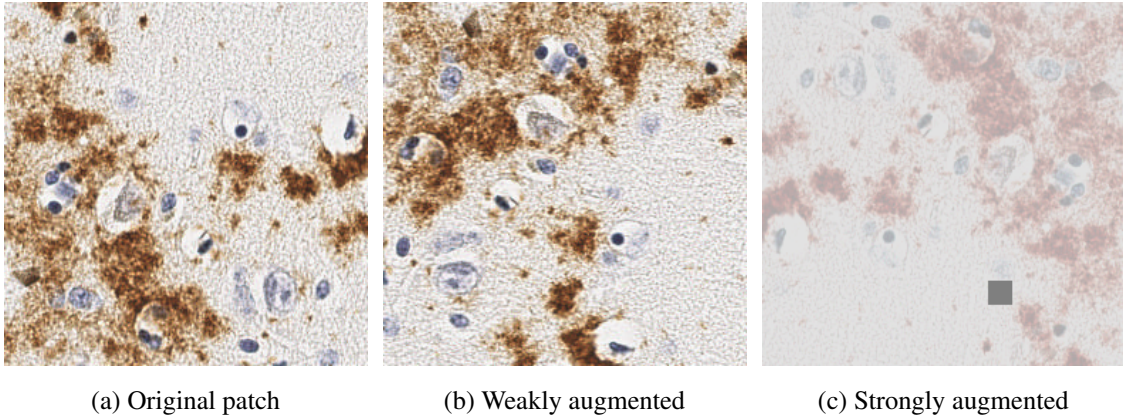


Figure 5.1. An example of weak/strong augmentation.

### 5.2.2 Semi-supervised active learning

One of the critical issues in applying SSL is the selection of the labeled data  $\mathcal{X}$ . In this subsection, we aim for an efficient selection of the labeled data by combining SSL and AL. The selection criteria of traditional AL [191, 193, 197] were designed based on only one sample in each cycle



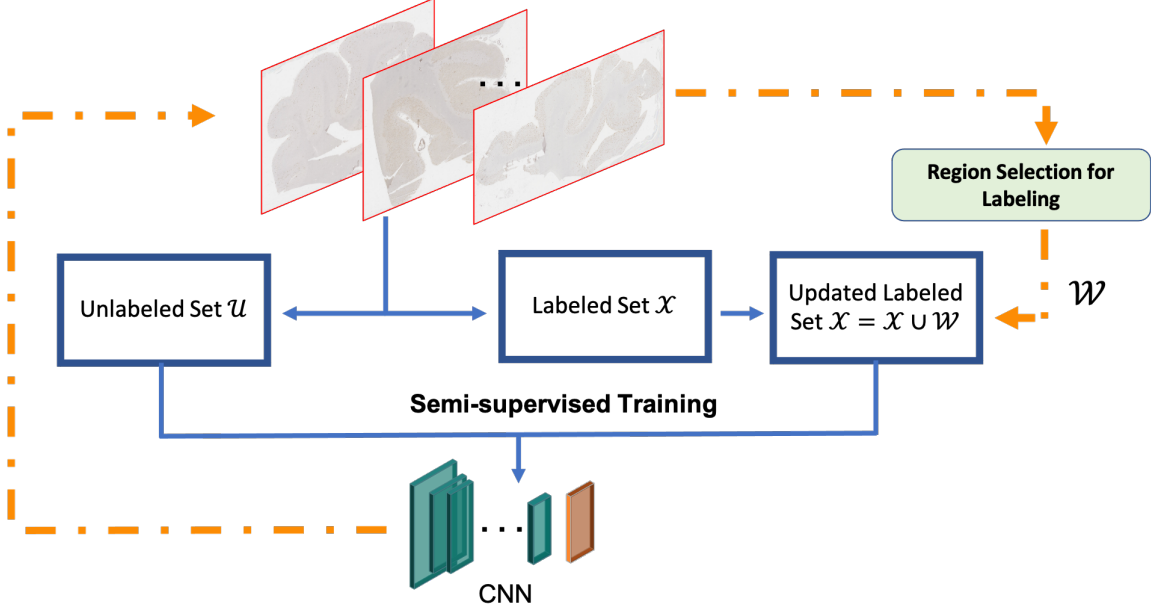


Figure 5.2. Overview of the proposed semi-supervised active learning framework for gigapixel pathology images.

of AL. Here one sample corresponds to one patch in our pathology images. However, one patch may not contain sufficient features for GM/WM and the criterion focusing on only one patch can lead to a haphazard selection for labeling query. Therefore, instead of patch-based selection, we propose a region-based selection criterion to not only reduce the number of labeling queries, but also to provide more “informative” data (details as shown in Session 5.2.3). Here a region  $\mathcal{R} \in \mathbb{R}^{nd \times nd}$  can be tiled into  $n^2$  patches. With the region-based selection, the labeled data set can be quickly expanded as

$$\mathcal{X} = \mathcal{X} \cup (\mathcal{W}, J(\mathcal{W})),$$

where  $\mathcal{W}$  represents a set of patches from  $\mathcal{R}$ .  $J(x)$  is the assigned label for  $x$ ,  $J(\mathcal{W})$  is a set of labels  $\{J(x)\}_{x \in \mathcal{W}}$ . The proposed framework is shown in Figure 5.2: starting from all WSIs, we deploy our proposed region-based selection criterion (details as shown in Session 5.2.3) to choose regions for annotation query. Then these regions would be added into  $\mathcal{X}$ , subsequently the updated  $\mathcal{X}$  and  $\mathcal{U}$  are used for semi-supervised training to optimize the encoder.

For a well-defined initialization, we also adopt a pre-training process by using a self-supervised module to pretrain the encoder on the unlabeled set  $\mathcal{U}$ . SimCLR [37] is a simple self-supervised framework that applies contrastive learning to capture visual representations on

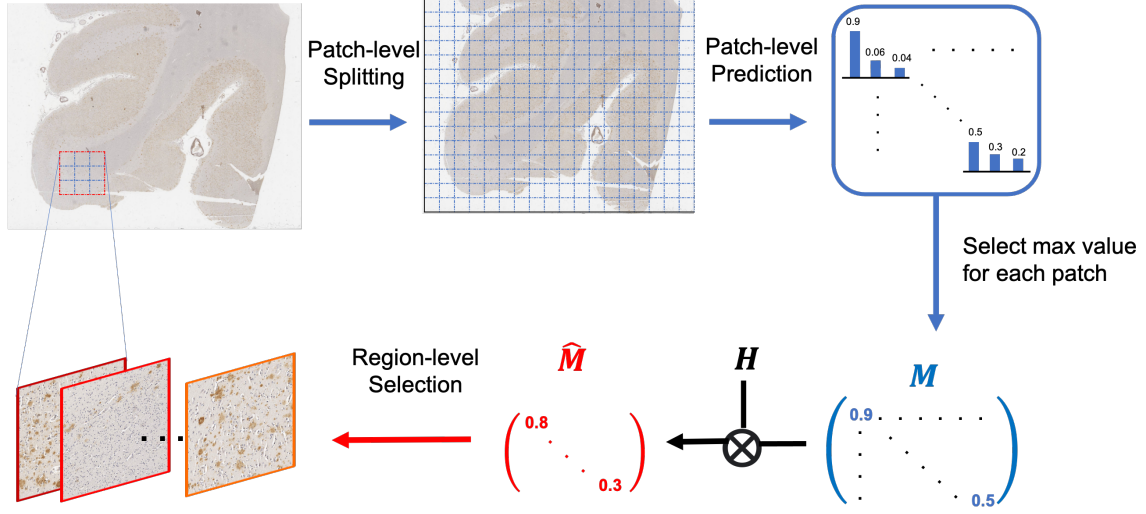


Figure 5.3. Illustration of the proposed region-based selection criterion: a sliding window is used to traverse the whole slide with semi-supervised learning model to produce patch-level prediction matrix  $M$ , which is then convoluted with kernel  $H$  to transform it into  $\hat{M}$  that contains region-level metrics.

unlabeled set. We adapt SimCLR [37] to train the encoder  $f$ . Then we select several diverse regions among WSIs for annotation query and fine-tune  $h = f \circ g$ , where  $g$  is the last layer for the classifier  $h$  as the **initialization** of our AL cycles.

### 5.2.3 Region-based selection criterion

In this subsection, we propose a region-based selection criterion for our AL framework. We hypothesize that labeling samples with high entropy in the prediction should be valuable. However, traditional uncertainty-based methods focus on each individual sample, which may result in haphazard and the “overconfident” issue [200]. This may lead the model to select less valuable samples. Therefore, to enhance the robustness of the uncertainty criterion, we propose a region-based selection criterion to focus on a region consisting of a batch of neighboring samples together instead of on only one sample at once. The regions where the majority of samples have high uncertainty would be selected for annotation query. The details of this process is shown in Figure 5.3.

Specifically, the proposed metric quantifies the accumulative entropy of predictions over a whole region: for each patch in the region, the value of the most confident class would be

---

**Algorithm 5** Region-based semi-supervised active learning for gigapixel histology images

---

**Input:** training dataset  $\mathcal{D}$ , the total number of steps and cycles  $S, T$ , a kernel  $H$ , the number of regions selected in one cycle  $m$

**Pre-training:** obtain  $f$  via SimCLR [37] among  $\mathcal{D}$

**Initialize:** randomly select two regions for pixel-wise labeling query and tile them into patches to formulate a labeled dataset  $\mathcal{X}_0$

Set an unlabeled data by  $\mathcal{U}_0 = \mathcal{D} / \mathcal{X}_0$

**Fine-tuning:**  $\theta_0 = \arg \min_{\theta \in \Theta} L(\mathcal{X}_0, \theta) + \Omega(\mathcal{D}, \theta)$

**while**  $t < T$  **do**

**for**  $s = 0, \dots, S - 1$  **do**

        Patch prediction on  $\mathcal{U}_t$

        Generate  $\hat{M}_s$  with the  $i$ -th entry being

$$\max\{h(x_i)_1, h(x_i)_2, h(x_i)_3\}$$

        Set those entries in  $M_s$  corresponding to regions from  $\mathcal{X}_t$  as infinity

$$\hat{M}_s = M_s * H$$

        Select the regions  $\{\mathcal{R}_i\}_{i=1}^m$  corresponding to the  $m$  lowest value of entries in  $\hat{M}_s$

        Split  $\{\mathcal{R}_i\}_{i=1}^r$  into a group of patches  $\mathcal{W}_s$

$$\mathcal{X}_t = \mathcal{X}_t \cup (\mathcal{W}_s, J(\mathcal{W}_s))$$

$$\mathcal{U}_t = \mathcal{U}_t / \mathcal{W}_s$$

**end for**

    Update  $\theta : \theta_{t+1} = \arg \min_{\theta \in \Theta} L(\mathcal{X}_t, \theta) + \Omega(\mathcal{D}, \theta)$

$t = t + 1$

**end while**

---

selected. Then we can generate a matrix  $M$  where the  $i$ -th entry is

$$m_i = \max\{h(x_i)_1, h(x_i)_2, h(x_i)_3\}, \quad (5.2)$$

As the classifier  $h$  is at the patch level, we need to transform the metric from the patch level into the region level. By introducing a kernel  $H$ , we process a convolution between  $M$  and  $H$ , i.e., as  $\hat{M} = M * H$ . Here  $H \in \mathbb{R}^{q \times q}$  could be a mean filter with all the entries being  $1/q^2$  to collect the

statistic information in each region. The dimension of  $H$  is determined based on the original resolution of the histology images: the size of  $H$  should be large enough to contain more areas with sufficient information from each class. In our case, as the boundaries of GM and WM form gradual changes across hundreds of pixels and the size of one patch is set as  $256 \times 256$ , we set the dimension of  $H$  as  $5 \times 5$ , which represents a corresponding region of  $1280 \times 1280$  pixels to include multiple classes in one region. Based on our hypothesis, we will select  $m$  regions with  $m$  lowest values (regions with most uncertainty) among entries of  $\hat{M}$ . To avoid redundant selection, the value of entries selected in the previous iterations will be set as infinity. We summarize the whole process in Algorithm 5.

## 5.3 Preliminary results

### 5.3.1 Training setup

We used the ResNet-18 [55] as the encoder for all experiments. The batch size is 32 for both labeled and unlabeled data. For the training optimizer, we use Adam with  $\beta_1 = 0.9$  and  $\beta_2 = 0.999$  in our experiments. The learning rate is 0.001. All computations were run on an Intel Xeon W-2102 CPU (4 cores / 4 threads) and a NVIDIA Titan Xp GPU with 12 GB of VRAM. Other hyper-parameters used in FixMatch are summarized in Table 5.1. As downsampling [14] would lose minute pathological features, we follow a previously published patch based method [15] to tile WSIs into  $256 \times 256$  patches. 20 WSIs (12 AD cases and 8 NAD cases) are selected for the training process inclusive of the training and validation set while 10 WSIs (6 AD cases and 4 NAD cases) are remained as the **hold-out test** set. We report the results of the **hold-out test** set in the following sections.

Table 5.1: FixMatch settings.

| Hyper-parameter                                  | Value |
|--------------------------------------------------|-------|
| Confidence threshold $\tau$                      | 0.95  |
| Unlabeled loss weight $\lambda_u$                | 1     |
| Ratio of unlabeled data in each mini-batch $\mu$ | 2     |

Table 5.2: Pixel-wise IoU Scores for AD, NAD, and overall test set

| Method       | FCN [17]         |                  | U-Net [27]       |                         | FixMatch [17]    | <b>Proposed AL-600/AL-400</b> |                         |
|--------------|------------------|------------------|------------------|-------------------------|------------------|-------------------------------|-------------------------|
| Labeled data | 2 WSIs           | All WSIs         | 2 WSIs           | All WSIs                | <b>0.1%</b>      | <b>0.1%</b>                   | <b>0.07%</b>            |
| AD Back      | 61.04 $\pm$ 5.44 | 81.13 $\pm$ 9.17 | 59.74 $\pm$ 13.9 | <b>96.80</b> $\pm$ 1.48 | 93.15 $\pm$ 2.41 | 95.01 $\pm$ 1.17              | <b>96.40</b> $\pm$ 0.95 |
| AD GM        | 46.98 $\pm$ 2.78 | 76.07 $\pm$ 8.91 | 37.16 $\pm$ 9.93 | <b>89.58</b> $\pm$ 5.12 | 78.57 $\pm$ 3.87 | 88.80 $\pm$ 3.92              | <b>89.71</b> $\pm$ 3.34 |
| AD WM        | 27.75 $\pm$ 5.50 | 62.23 $\pm$ 14.0 | 7.57 $\pm$ 6.02  | <b>82.53</b> $\pm$ 7.70 | 56.66 $\pm$ 16.4 | 81.83 $\pm$ 5.53              | <b>82.19</b> $\pm$ 3.77 |
| AD Mean      | 45.26 $\pm$ 3.55 | 73.14 $\pm$ 9.66 | 35.40 $\pm$ 7.12 | <b>89.64</b> $\pm$ 4.35 | 76.13 $\pm$ 5.89 | 88.55 $\pm$ 3.27              | <b>89.44</b> $\pm$ 2.50 |
| NAD Back     | 66.66 $\pm$ 5.17 | 88.42 $\pm$ 1.55 | 78.46 $\pm$ 18.5 | <b>97.36</b> $\pm$ 3.15 | 97.07 $\pm$ 0.31 | 97.26 $\pm$ 0.52              | <b>97.33</b> $\pm$ 1.05 |
| NAD GM       | 50.15 $\pm$ 0.49 | 79.37 $\pm$ 2.95 | 59.59 $\pm$ 13.6 | <b>94.42</b> $\pm$ 3.30 | 83.97 $\pm$ 7.76 | <b>93.47</b> $\pm$ 1.60       | 92.25 $\pm$ 2.36        |
| NAD WM       | 19.72 $\pm$ 13.6 | 49.89 $\pm$ 12.8 | 3.02 $\pm$ 3.09  | <b>81.25</b> $\pm$ 9.53 | 22.72 $\pm$ 19.0 | <b>75.85</b> $\pm$ 11.4       | 71.90 $\pm$ 11.2        |
| NAD Mean     | 45.51 $\pm$ 3.29 | 72.56 $\pm$ 3.97 | 47.02 $\pm$ 10.9 | <b>91.01</b> $\pm$ 3.36 | 67.92 $\pm$ 6.53 | <b>88.86</b> $\pm$ 3.38       | 87.16 $\pm$ 2.86        |
| Test Back    | 63.29 $\pm$ 5.81 | 84.05 $\pm$ 9.17 | 68.28 $\pm$ 17.2 | <b>97.02</b> $\pm$ 2.15 | 94.72 $\pm$ 2.71 | 95.91 $\pm$ 1.48              | <b>96.77</b> $\pm$ 1.05 |
| Test GM      | 48.25 $\pm$ 2.66 | 77.39 $\pm$ 7.06 | 46.13 $\pm$ 15.8 | <b>91.52</b> $\pm$ 4.94 | 80.73 $\pm$ 6.01 | 90.67 $\pm$ 3.90              | <b>90.73</b> $\pm$ 3.13 |
| Test WM      | 24.54 $\pm$ 9.80 | 57.29 $\pm$ 14.3 | 5.75 $\pm$ 5.37  | <b>82.02</b> $\pm$ 7.98 | 43.08 $\pm$ 24.0 | <b>79.44</b> $\pm$ 8.34       | 78.07 $\pm$ 8.81        |
| Test Mean    | 45.36 $\pm$ 3.26 | 72.91 $\pm$ 7.56 | 40.05 $\pm$ 10.2 | <b>90.19</b> $\pm$ 3.84 | 72.84 $\pm$ 7.18 | <b>88.67</b> $\pm$ 3.12       | 88.53 $\pm$ 2.75        |

Rows marked AD contain results on the 6 Alzheimer’s disease cases in *hold-out* test set. Rows marked NAD contain results on the 4 non-Alzheimer’s disease cases in test set. Rows marked Test contain results on all 10 WSIs. 2 WSIs refers to 2 WSIs are labeled, equivalent to 10% regions of all WSIs; all WSIs refers to all WSIs are labeled. 0.1% refers to 0.1% regions of all WSIs are labeled, which can be tiled into 600 patches; so as 0.07% which can be tiled into 400 patches.

### 5.3.2 Quantitative comparison

Since our target task is GM/WM segmentation, we choose two standard segmentation metrics — IoU score [38] and DICE coefficient [46] to compare the generated masks from different approaches with the ground truth masks.

**IoU score.** IoU score (as shown in (5.3)) measures the amount of overlap present in two different measurements of area generated as masks of the WSIs.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}. \quad (5.3)$$

Although IoU is the ideal metric for evaluating the success of the segmentation, average IoUs cannot account for the variability that occur between different methods. This variability is important to account for as the different Temporal Gyri WSIs contain noticeable differences due to the heterogeneous nature of the human brain. Hence, we choose to also report standard deviation (STD) in Table 5.2 to measure the stability and consistency of different methods across multiple WSIs in the hold-out test set. We generate the masks of GM, WM, and background per WSI from the trained model and overlap them on pixel-wise ground truth masks. The comparison of IoU scores and STD are summarized in Table 5.2.

As shown in Table 5.2, if FCN [11] and U-Net [27] are trained on *all* WSIs with annotations in the training set via fully-supervised learning, they are able to achieve the mean IoU score at nearly 72.91% and 90.19%, respectively. 90.19% would be approximately regarded as an upper bound performance of supervised learning (SL) in this dataset. However, if they are only trained on two annotated WSIs (one AD and one NAD) with SL, the performance has a severe degradation.

After we obtain the benchmark of SL algorithms, we first deploy the current state-of-the-art SSL algorithm, FixMatch [17], on our dataset with the same two WSIs annotated in pixel level. As it shows its advantages on very scarce limited data, e.g. 40 labeled in CIFAR-10, equivalent to nearly 0.1% of all samples in CIFAR-10 [17], we follow their way to set 0.1% of labeled data as the objective. In our case, 0.1% area of 20 WSIs is about 600 patches, equivalent to 24 regions at  $1280 \times 1280$  pixels. We first adopt a *randomly uniform* selection to select 600 patches from two WSIs (300 patches per WSI, and 100 patches per class). The remaining patches and 18 slides keep as the unlabeled set. It achieves 72.84% of mean IoU score on the hold-out test set, which is superior to FCN/U-Net’s performance when trained only on two WSIs (limited labeled data). However, its performance on the WM of NAD cases remains limited and far from SL’s.

Then we evaluate our proposed framework with region-based selection cycles. To obtain 0.1% area (or 24 regions tiled into 600 patches), we use AL-400 to refer to our proposed method

that selects 16 regions at  $1280 \times 1280$  pixels in two rounds to generate 400 patches in total for preparing the labeled set of FixMatch [17]. While AL-600 refers to selection of 24 regions in three rounds to generate 600 patches in total for labeling. AL-400 is able to reach almost 89.44% of mean IoU in AD cases, which is very close to the upper bound of SL when trained on all labeled WSI data. However, there is an notable gap between its IoU score on NAD cases compared to U-Net, especially in WM where the gap is almost 10%. After an addition round for region selection and annotation query, AL-600 improves its performance on NAD cases with nearly 4% of increase on the IoU score of WM. One noticeable issue is that the STD values of WM on NAD cases are relatively larger than the values on other regions, which indicates that the model still has difficulty in predicting WM in NAD slides. This may be caused by the imbalance between WM and GM in NAD cases: the size of WM regions is relatively smaller than that of GM regions. Overall, both AL-400 and AL-600 have potential to get closer to the bound of SL.

**DICE coefficient.** Besides the IoU score, we also use DICE coefficient [129] to further compare our proposed approach with baseline FixMatch [17]. DICE is one of the most common methods of evaluating image segmentation success in medical imaging [46]. The value is calculated as shown in (5.3): double the overlapping areas and divide it by the total number of pixels in all areas. As shown in Figure 5.4a, the proposed methods outperform significantly on WM regions on DICE coefficient and also slightly outperform on GM regions compared to the baseline FixMatch [17] with random selection for labeling query.

$$\text{DICE} = \frac{2 \times \text{Area of Overlap}}{\text{Total Sum of Pixels in All Areas}}. \quad (5.4)$$

### 5.3.3 Segmentation visualization

Figure 5.5 display the ground truth masks and the predictive masks using different methods. The top two rows are the masks for AD cases while the bottom two rows are for NAD cases from the hold-out test set. The masks of U-Net(in the second column) indicate that SL is able to get well-defined performance as long as the variety and volume of labeled data are sufficiently large, but the performance could be degraded severely if the variety and volume of labeled data are limited (as shown in the third column). Baseline FixMatch with random selection for the labeled set (in the fourth column) is able to locate the rough boundaries between GM and WM but the

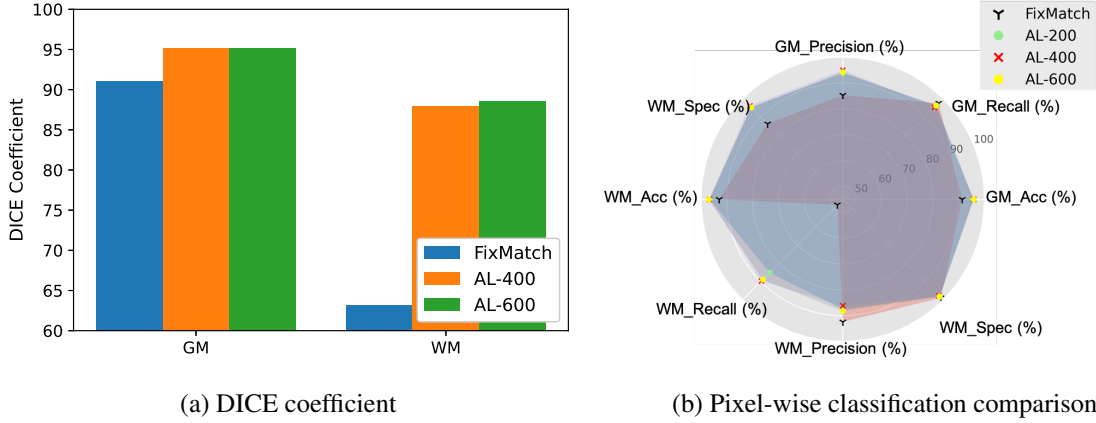


Figure 5.4. DICE coefficient and pixel-wise classification comparison via each AL cycle)

shape of WM is far wary from the ground truth masks. There are also considerable amounts of noisy pixels within WM, which means it wrongly classifies some WM regions as GM. Our proposed method with AL selection (in the fifth column) is able to provide more distinguishable boundaries for GM and WM regions. The generated masks are also the closest to our ground truth masks. However, for the second row, the masks of both FixMatch and ours are different from the ground truth masks, which indicates that SSL still requires more labeled data for in order to fully capture the representations of gigapixel histology images.

### 5.3.4 A deeper look into AL cycles

To further analyze the effect of AL cycles on FixMatch [17], the differences of IoU scores in each round are trivial. Therefore, we also select Accuracy, Recall (Sensitivity), Specificity, and Precision to have a deeper look into the AL’s effectiveness. These metrics are defined as follow:

$$\begin{aligned}
 \text{Accuracy} &= \frac{TP + TN}{TP + TN + FP + FN}, \\
 \text{Recall} &= \frac{TP}{TP + FN}, \\
 \text{Specificity} &= \frac{TN}{TN + FP}, \\
 \text{Precision} &= \frac{TP}{TP + FP}.
 \end{aligned} \tag{5.5}$$

Take GM as the example of how to calculate them: the True Positive (TP) of GM is all GM pixels that are classified as GM; the True Negative (TN) of GM is all non-GM pixels that are not classified as GM; the False Positive (FP) is all non-GM pixels that are classified as GM; the



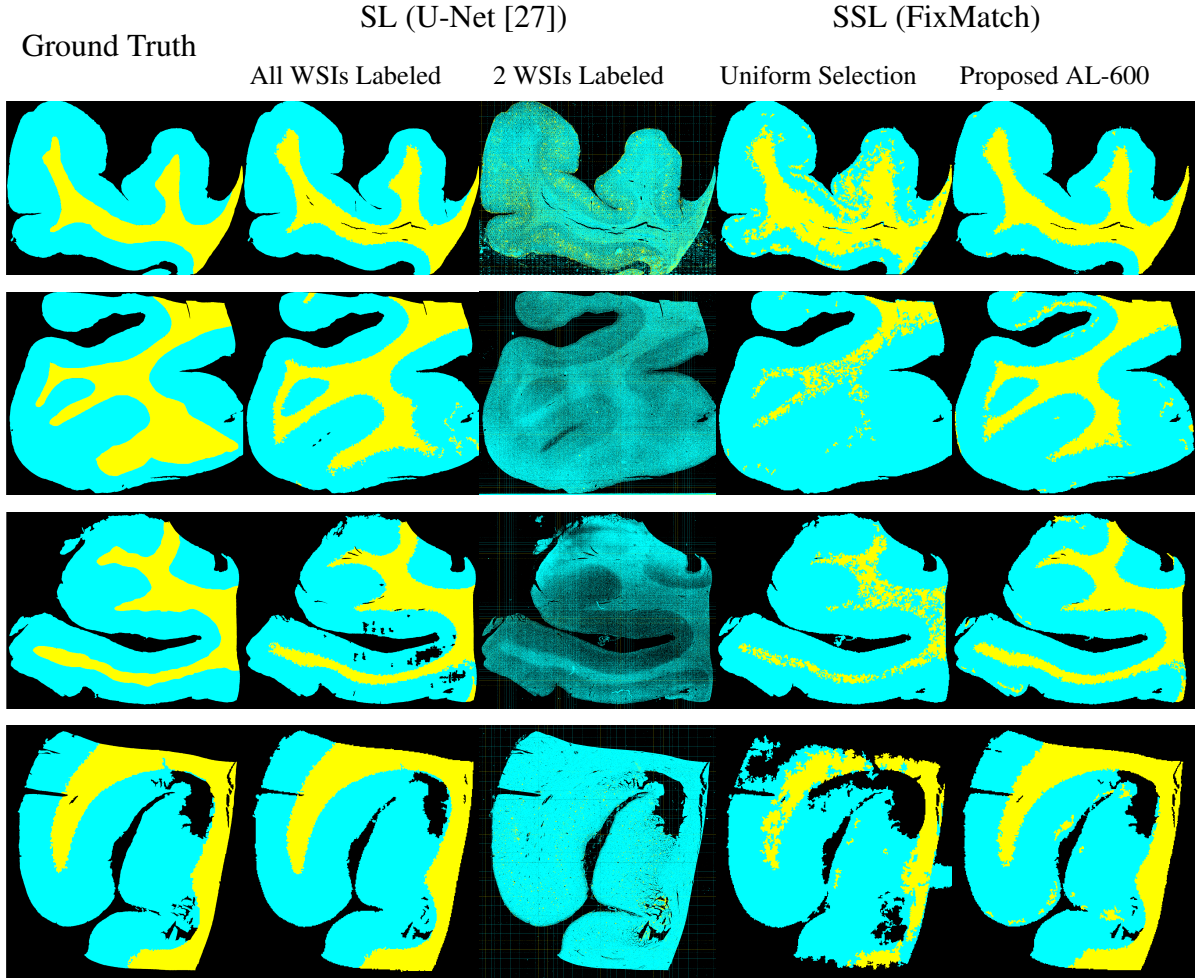


Figure 5.5. Segmentation masks visualization: GM, WM, and background are indicated by cyan, yellow, and black, respectively. The top two rows are AD cases and the bottom two rows are NAD cases. Both SSL results are only using 0.1% area of 20 WSIs in the training set.

False Negative (FN) is all GM pixels that are not classified as GM. The results for each AL cycle on the hold-out test set are summarized in Figure 5.4b. Our proposed framework shows higher precision and specificity in GM regions compared to original FixMatch [17]. It is obvious that WM’s recall of FixMatch [17] is the lowest score among all indices. This indicates our proposed framework is able to improve the performance on WM regions.

When we take a deeper look into the performance via AL cycles, from AL-200, AL-400, to AL-600, the model improves its performance on WM regions, such as recall. This fits our design objective: our framework always queries the annotations on difficult regions.

### 5.3.5 Analysis on number of regions selected

As stated in Algorithm 5,  $m$  is the number of regions selected for annotation query in one round of AL. In Section 5.3.2 and 5.3.3, we use three rounds to obtain 24 labeled regions, which are tiled into 600 patches in total. To further reduce the number of cycles, we also try to obtain the same number of regions at once based on  $f$  at once. We achieve nearly 87% of mean IoU score, which is only around 2% lower than AL-600 using three rounds of labeling query. This shows a promising direction on further studying the number rounds of labeling query that is required in the histology images.

## 5.4 Discussion

In this section, we investigate the applicability of supervised and semi-supervised learning state-of-the-art algorithms in histology images with limited labeled data. We propose a semi-supervised active learning framework with a region-based selection criteria to effectively select regions for labeling query. For our specific dataset, the region-based criterion is more robust than patch-based criterion in seeking the uncertainty regions for labeling and is able to quickly expand the diversity of the labeled set. We evaluate our framework on Amyloid- $\beta$  stained neuropathology images: our proposed algorithm outperforms the state-of-the-art models with more than 10% of IoU score and DICE coefficient, especially for the WM.

As part of our future work, we plan to test our framework on external datasets, as well as other ultra-resolution histology problems with scarce labeled data and comprehensively measure how much manual annotation efforts can be saved. Besides, considering the fact that we only explore one selection criterion for one specific WSI dataset, we plan to comprehensively study the selection criteria for determining a general rule for pathology images about which regions should be labeled first and how many data are sufficient for labeling. On the other hand, our active learning module assumes that we can interact with annotators on time. This can be impractical in real-world settings. Furthermore, even when we have such online annotators, the active learning module still requires many rounds of labeling queries, which is time-consuming [203]. As such, one-round or few-round active learning can be a boost in pathology image analysis. A recent study [203] proposed DULO to try to tackle the one-round issue based on data utility function

maximization. However, its applicability and adaptation on pathology images are still under study. This will be one of our future directions.

# Chapter 6

## A Deployable Semi-supervised Learning with Informative Active Annotation for Pathology Image Classification and Segmentation

Image analysis in digital pathology presents significant challenges for AI-driven classification, segmentation, and object detection due to the heterogeneity and ultra-high-resolution characteristics of pathology images. Manual annotation of these gigapixel-level images is time-consuming and labor-intensive. Semi-supervised learning (SSL) has emerged as a promising approach that leverages unlabeled data to reduce labeling efforts. In this work, we introduce Semi-Path, a practical SSL framework enhanced with active learning (AL) for gigapixel pathology tasks. Unlike existing methods that treat SSL and AL as independent components where AL incurs significant computational complexity to SSL, we propose a deep fusion of SSL and AL into a unified framework. Our framework introduces Informative Active Annotation (IAA) that employs an inner-outer two-loop structure to effectively extract knowledge from unlabeled pathology data. This structure significantly minimizes labeling efforts and computational complexity. Then, we propose Adaptive Pseudo Labeling (APL) to address heterogeneity in class distribution and prediction difficulty that are often observed in real-world pathology tasks. We evaluate Semi-Path on pathology image classification and segmentation tasks over three datasets that include WSIs

from breast, colorectal, and brain tissues) and show the consistent superiority of Semi-Path over state-of-the-art methods.

## 6.1 Introduction

Deep learning frameworks have shown exceptional performance on diverse imaging modalities in pathology [204, 205, 206]. For example, ResNet-based [55] networks can achieve superior results in pathology classification problems [15] while FCN [11] and U-Net [27] are two popular deep-learning frameworks that produce excellent segmentation results of pathology images. However, the performance of these supervised deep learning methods heavily rely on a large-scale and well-curated labeled dataset [204, 205, 206]. Annotating such a dataset can be a labor-intensive and time-consuming task considering the heterogeneous nature of pathology images and their ultra-high spatial resolutions (often up to  $60,000 \times 50,000$  pixels [72]).

Semi-supervised learning (SSL) can leverage unlabeled data to reduce the labeling efforts and has shown promising results in computer vision tasks [17, 18, 19, 20]. However, the efforts to deploy them on gigapixel pathology tasks remain limited [21, 22, 23, 24]. For example, a recent work [21] found the performance of FixMatch [17] on a pathology dataset is unsatisfactory and sensitive to patient diversity. There are two possible reasons why current SSL methods are ineffective. The first reason is that unlabeled data may not be fully utilized by the learning algorithms: pathology images of ultra-high resolution can be highly similar at the pixel level [71], making them hard to distinguish, even by trained researchers. This challenging characteristic harms the performance of the recent SSL algorithms from computer vision tasks in the early stage of the training process. Second, these SSL algorithms may miss informative samples: the samples not selected for pseudo-labeling (whose prediction confidence is below the pre-defined threshold [17, 148]) may have extra information useful for learning, e.g., uncertain samples. Based on these two reasons, we propose building an interactive SSL framework to deeply and efficiently excavate relevant information from unlabeled data for gigapixel pathology image tasks.

To deeply exploit the informativeness of the unlabeled data, we propose class-aware Adaptive Pseudo Labeling (APL) to dynamically select the samples of the unlabeled pool for pseudo-

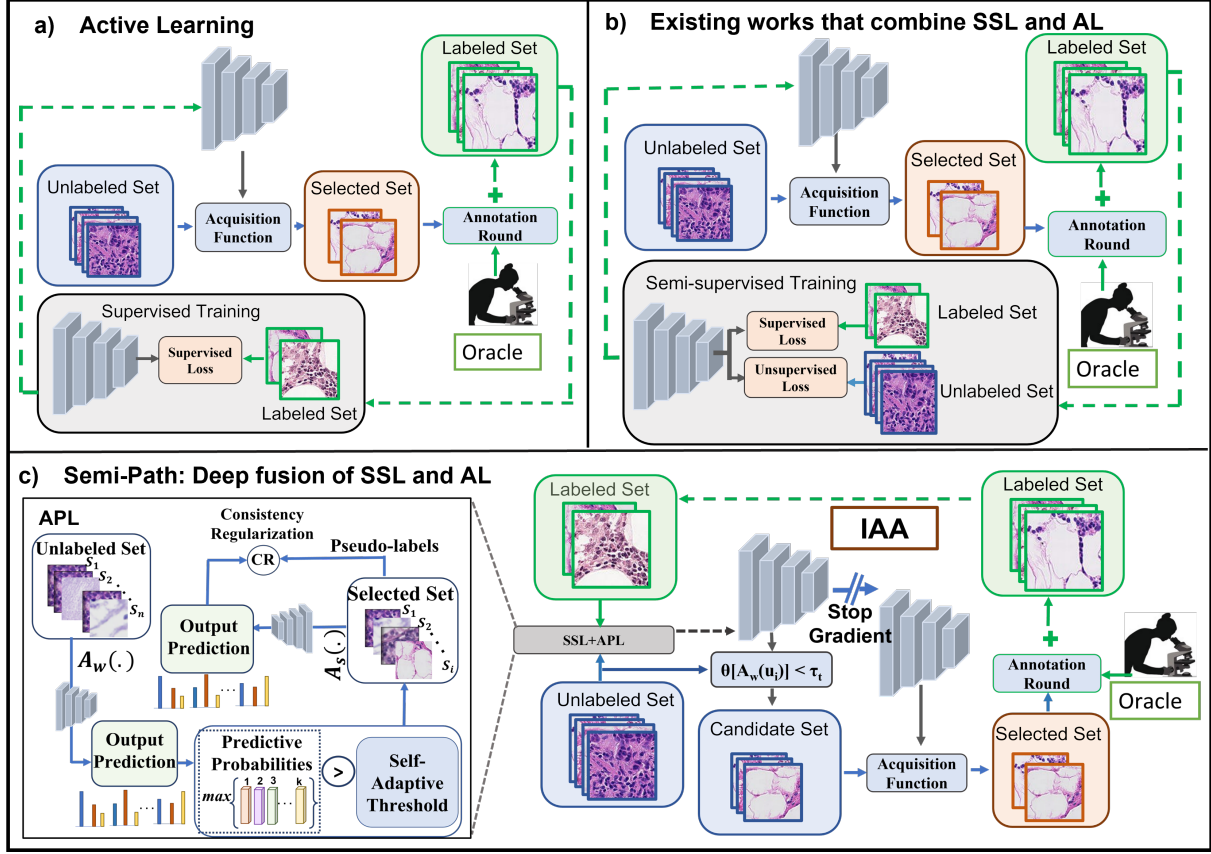


Figure 6.1. The difference between Semi-Path and existing methods that combine SSL and AL: instead of treating them as independent components, we propose deep fusing them with an inner-outer two-loop structure. APL is proposed to exploit the unlabeled data during SSL training further.

labeling based on the learning status of each class in the SSL training process. In addition, we incorporate active learning (AL) to select the informative samples to construct the labeled set for SSL training. As shown in Fig. 6.1b, existing works [72, 200] that combine SSL and AL treat SSL and AL as separate and independent steps, which will incur heavy computational complexity from AL on gigapixel pathology images. Different from these methods, we deeply delve into SSL training and propose a deep fusion of SSL and AL to avoid additional computational complexity incurred by AL. To achieve such a deep fusion, we propose an inner-outer two-loop structure to exploit the relationship between SSL and AL. We show that the SSL training can serve a similar function to the inference process in AL so we adapt SSL training as the inner-loop selection to identify a candidate pool for AL without incurring extra computational cost. Our

main contributions are summarized as follows.

- We explore the applicability of SSL on gigapixel-level pathology images and propose Semi-Path to minimize the labeling efforts.
- To effectively excavate the unlabeled data, we propose class-aware Adaptive Pseudo Labeling (APL) to mitigate the heterogeneity issue of pathology slides.
- We propose Informative Active Annotation (IAA) to enhance SSL with AL in a deep fusion achieved by the proposed inner-outer two-loop structure.
- We show that Semi-Path achieves consistently superior performance over state-of-the-art SSL approaches across three datasets and two learning tasks.

## 6.2 Related work

**Semi-supervised learning (SSL).** Recent SSL methods utilize pseudo labeling, consistency regularization, and/or a combination of them [72, 207] to train deep neural networks with both labeled and unlabeled data. Pseudo-labeling methods [147, 141] use the intermediate models to produce pseudo labels for some of the unlabeled data, which are then combined with the labeled set to train the model further. Consistency-based methods [151, 140] regularize the model training by encouraging consistent predictions on the same sample under various perturbations. Recent works [17, 18, 19, 20] combine these two approaches and achieve better performance. Although these SSL algorithms have shown promising results in computer vision tasks, their applicability on gigapixel pathology tasks has not been sufficiently explored and verified: Pulido *et al.* [21] found that the performance of SSL methods can be adversely affected by patient diversity; recent works [207, 208, 209] also observed that there is still a significant performance gap between fully-supervised learning and SSL. Specifically, there are two challenges for these SSL methods: 1) poor-quality pseudo labels may result in self-reinforcing errors during the training process [34]; 2) insufficiency in mining the potential of unlabeled data [72].

**Active learning (AL).** Most AL methods choose informative samples by optimizing an acquisition function, which can be defined based on uncertainty, diversity, change in model performance, or a combination of these metrics. The uncertainty-based AL methods use *max entropy*

or *max margin* [190] as the selection criteria due to their simplicity and effectiveness [188]. To reduce the number of AL cycles, batch-mode AL methods [181, 182, 183, 185] were proposed to select a batch of data at once instead of a single data point for labeling. All these AL methods need to perform, for each AL cycle, inference on each unlabeled sample and optimization of the target acquisition function, which can be computationally intensive [210]. The computational challenge can be greatly exacerbated when applied to gigapixel pathological images. For example, it took over one hour with one Nvidia Titan Xp to make the inference on one gigapixel WSI [72].

**Semi-supervised active learning (SSAL).** Although SSL and AL have a similar goal of reducing labeling costs, only a few studies have attempted to combine them into a unified framework. Drugman *et al.* combined SSL and AL for speech understanding under limited speech data [198]. Rhee *et al.* combined SSL and AL for a pedestrian detection task [199]. Mahapatra *et al.* [211] combined SSL and AL for segmenting MR images. Although these methods attempt to combine SSL and AL, their selection procedures in AL cycles are independent of the model training. Specifically, SSL training and AL selection are two separate steps in the above methods. In addition, none of them have been evaluated on the gigapixel pathology settings.

## 6.3 Methodology

Before presenting our framework, we first introduce the basic notions and terminologies. In semi-supervision settings, the training set includes a labeled data pool  $\mathcal{X}$  with the corresponding labels  $\mathcal{Y}$  and an unlabeled data pool  $\mathcal{U}$ . The goal is to use both  $\mathcal{X}$  and  $\mathcal{U}$  to train the model in a semi-supervised manner. We denote the predictive probability from the model as  $h(x; \theta)$ , where  $\theta$  refers to the model. We use  $N_l$  as the number of images in the labeled pool  $\mathcal{X}$ . Given labeled data  $\mathcal{X} = \{x_i\}_{i=1}^{N_l}$  and their labels  $\mathcal{Y} = \{y_i\}_{i=1}^{N_l}$ , we apply weak augmentations denoted as  $A_w(\cdot)$  (using only flip-and-shift) and minimize the cross-entropy loss through supervised label information as  $\mathcal{L}(x, y, \theta) = D_{ce}(y, h(A_w(x); \theta))$ , where  $D_{ce}(\cdot, \cdot)$  denotes the cross-entropy between the predictive output and the label. Similar to many recent state-of-the-art SSL algorithms [17, 19, 20], we have supervised and unsupervised loss together to train the model  $\theta$  via the following



optimization:

$$\min_{\theta \in \Theta} \Omega(\mathcal{U}, \theta) + \frac{\alpha}{N_l} \sum_{\substack{x \in \mathcal{X} \\ y \in \mathcal{Y}}} \mathcal{L}(x, \mathbf{y}, \theta), \quad (6.1)$$

where  $\alpha$  is a hyperparameter that balances the supervised and consistency loss, and  $\Omega$  refers to the unsupervised loss on the unlabeled data (see in the next section).

### 6.3.1 Class-aware Adaptive Pseudo Labeling (APL)

The pseudo label of an unlabeled image is generated based on the model’s prediction on the weakly-augmented version of the image when its confidence exceeds a pre-defined threshold [17]. The success of the pseudo-labeling with a fixed threshold relies on a necessary assumption that labeled and unlabeled data have similar distributions [212, 41]. This assumption often does not hold in real-life applications, especially in the pathology domain [21], and thus it could hurt the original model’s performance [31]. As a result, if the model generates the wrong pseudo labels due to the domain and distribution shift, and utilizes them as the supervision, the model will be biased and have the prediction error reinforced during the self-training process.

In this work, we propose an adaptive threshold  $\tau_t$  based on learning status and learning difficulty to address mismatched distributions in real-life applications. The criteria for selecting an unlabeled sample  $x$  for pseudo-labeling is as follows:

$$\max_k h(A_w(x); \theta)_k > \tau_t \quad (6.2)$$

The pseudo-label is then defined as a one-hot vector  $\mathbf{p}(x) = \arg \max_k h(A_w(x); \theta)_k$  if  $x$  satisfies the above criterion. The adaptive selection is critical in the semi-supervised settings: if  $\tau_t$  is too high, we may miss some informative samples from the unlabeled data for the supervision; if  $\tau_t$  is too low, we may incorporate many low-quality pseudo labels that may hurt the model’s performance. On the other hand, the learning status is also different in each training iteration. Hence we propose to design a dynamic threshold for each class based on its learning effect.

A recent SSL work [18] argued the learning status of the SSL model could be reflected by the number of unlabeled samples with predictions above the threshold. Based on this observation,

we design  $\tau_t$  per epoch as:

$$\tau_t = \begin{cases} \tau \cdot \min\{1, N_u^t/N_u^{t-1}\}, & \text{if } 1 < t \leq T_{max} \\ \tau, & \text{otherwise} \end{cases} \quad (6.3)$$

where  $t$  indicates the epoch number and  $N_u^t = \sum_{x \in \mathcal{U}} \mathbf{1}(\max_k h(A_w(x); \theta)_k > \tau_{t-1})$ , with  $\mathbf{1}(\cdot)$  being the indicator function, is to quantify the number of pseudo labels from last epoch. We follow FixMatch [17] to set the initial  $\tau_0$  as 0.95: when there are more samples selected out for pseudo-labeling,  $\tau_t$  will remain at a high level to reserve the most confident samples; otherwise,  $\tau_t$  can be lower to incorporate more samples for pseudo-labeling to encourage the better utilization of unlabeled data until it reaches the maximum iterations. This dynamic mechanism can adjust the potential bias from the distribution shift: a potential distribution shift may result in the model's over-confidence in the majority class and thus the pseudo labels on the minority classes can be under-estimated. With the class-aware self-adaptive pseudo-labeling, the model will balance its learning status and generate less-biased pseudo labels.

We now define the unsupervised loss in (6.1). It is the augmentation-aware consistency loss [17], where the model is trained to enforce the consistency between the predictions on the strongly-augmented unlabeled image  $A_s(x)$  and the pseudo labels  $\mathbf{p}(x)$  derived from the weakly-augmented version via a cross-entropy loss:

$$\Omega(\mathcal{U}, \theta) = \frac{1}{N_u^t} \sum_{x \in \mathcal{U}} \mathbf{1}(\max_k h(A_w(x); \theta)_k > \tau_{t-1}) \cdot D_{ce}(\mathbf{p}(x), h(A_s(x), \theta)). \quad (6.4)$$

### 6.3.2 Informative Active Annotation (IAA): Deep fusion of SSL and AL

Existing SSAL methods [72, 210, 200] utilized AL as an independent step and incurred extra computational cost. This makes it difficult to be deployed widely in pathology domains for two reasons: first, the computational burden can be extremely heavy when the acquisition function is complicated, e.g. non-linear optimization objective [213, 214, 215]; second, even with computationally-efficient acquisition functions, the gigapixel size of WSIs still poses a formidable computational challenge [216] to make AL inference on them. To deal with these challenges, we propose a novel method called Informative Active Annotation (IAA) to deeply

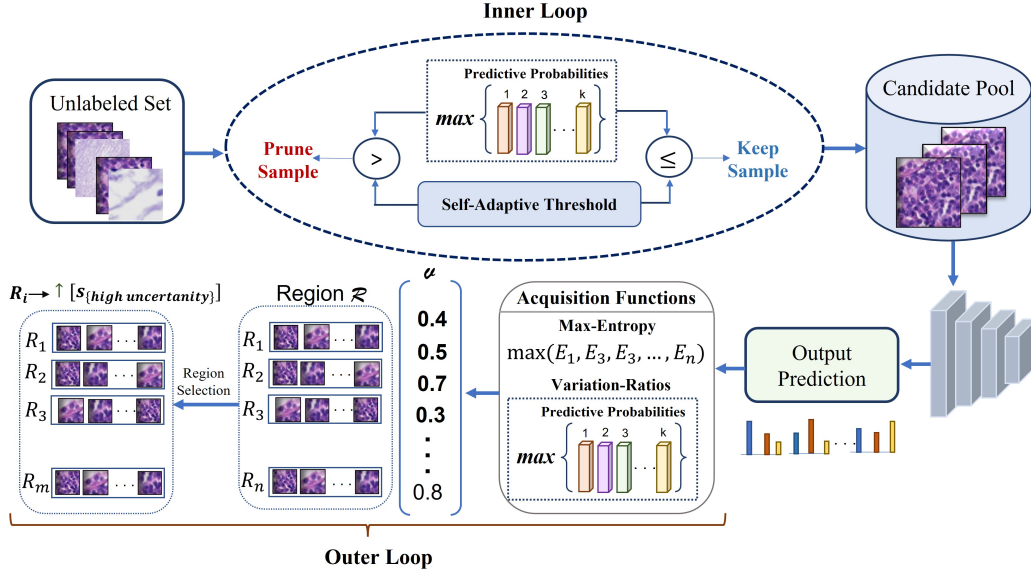


Figure 6.2. Architecture of the proposed Informative Active Annotation (IAA) module. SSL training performs the inner-loop pruning of the candidate pool for the outer loop, then the outer loop utilizes AL’s acquisition function to select the samples for labeling.

fuse SSL and AL. Fig. 6.2 illustrates the overall framework of our proposed IAA module. We design an inner-outer two-loop structure specifically to fuse SSL and AL to construct an informative labeled set without incurring additional heavy computational costs.

### 6.3.2.1 Inner-loop construction of candidate pool

Since the inference stage for the entire unlabeled set in AL involves extensive computations, especially in the gigapixel-resolution pathology domain, it is essential to explore the potential methods for reducing the dataset size for such inference.

The key objective for AL is to determine which samples with high uncertainty so that acquiring their labels can improve the model training process. In SSL, we can identify uncertain samples if we are unable to assign pseudo-labels to them. In other words, the samples with a maximum predictive probability below the threshold  $\tau_T$ , which is self-adapted in the last epoch of this round of SSL training, will remain in the candidate pool  $\mathcal{C}_r$ . We define  $\mathcal{C}_{r-1}$  as the candidate pool from last round of IAA with  $\mathcal{C}_0 = \mathcal{U}$  as the initial candidate pool. For each round of IAA, we can get

$$\mathcal{C}_r = \{x \in \mathcal{C}_{r-1} : \max_k h(A_w(x); \theta)_k > \tau_T\}. \quad (6.5)$$

Consequently, the pseudo-labeling process employed in SSL training can serve as an ac-

quisition function in AL. The most commonly used acquisition function is the entropy of the predictive probability [217, 218]. As such, it is important to show that uncertain samples selected with entropy will not be pruned away during inner-loop selection using pseudo-labeling. Focusing on binary classification, we prove that minimizing the inner-loop function is equivalent to maximizing the entropy thus our inner-loop pruning will not prune uncertain and informative samples.

Overall, our proposed SSL module segregates the confident samples and reconstructs the candidate pool of samples not selected for pseudo-labeling. Then the AL inference can be performed solely on this pruned candidate pool, saving computation time and resources by avoiding inference on the entire unlabeled set.

### 6.3.2.2 Outer-loop AL selection

We have shown SSL training process can serve as AL inference. After the pruning of the candidate pool, we step into the outer-loop of IAA, as shown in Fig. 6.2. Acquisition functions from AL are introduced in this step to rank samples to fit the labeling budget. If the labeling budget is larger than the size of the candidate pool, we will skip the outer-loop selection but annotate all samples in the pool with the labeling budget. If the labeling budget is smaller than the candidate pool, we will enter outer-loop AL selection: for each cycle, we annotate a subset of the pool until we reach the labeling effort. First, we calculate the score for each sample in the candidate pool and save them in a vector  $v \in \mathbb{R}^n$ , where  $n$  is the size of the candidate pool. Based on the labeling budget, we then select  $m$  samples with the lowest confidence among entries of  $v$ . However, it is important to define an AL selection criterion for selecting the most informative samples. One of the widely-used selection criteria is based on the degree of uncertainty the current model has on each unlabeled sample.

The gigapixel size of pathology images introduces further challenges in using the traditional AL methods. AL process itself requires high-end computational resources to finish the task. When these methods are used for high-resolution pathology images, it may add more complexity to the model and time delays to the pathology image analysis task. In addition, the slow training due to high computational complexity may lead to additional delays in synchronizing the annotator availability after each AL iteration. Considering the gigapixel resolution of

pathology images, we evaluate two computationally efficient yet effective methods in IAA: *Max-Entropy* [218] and *Variation-Ratios* [217, 218]. For *Max-Entropy* [218], samples with high entropy are considered informative and selected for labeling as higher entropy indicates higher uncertainty [219]. For *Variation-Ratios* [217, 218], we select the highest probability  $\max_k \{h(x, \theta)_k\}$  as the acquisition function in each patch’s soft label. This value can describe the degree of confidence and uncertainty.

### 6.3.3 Extension to gigapixel-level segmentation tasks

As pathology images are typically at the gigapixel level, annotating the entire WSIs can be more labor-intensive and time-consuming for the segmentation tasks. Here we show the capability of extending Semi-Path to weakly-supervised pathology image segmentation tasks: instead of requesting pixel-level annotation of an entire WSI, we focus on seeking the most informative regions within the WSI for the annotation. In [72], they verified a small set of “difficult” regions may be sufficient to improve the model’s performance since the visual representations can be similar in gigapixel slides. Therefore, we aim to seek those informative regions to further reduce the labeling efforts by avoiding labeling the whole slide in a weakly-supervised manner.

To enhance the robustness of the uncertainty criterion, we follow [72] to integrate a region-based selection criterion. Instead of focusing on only one sample at a time, it focuses on a region consisting of a batch of neighboring samples that capture more neighboring information. The regions where the majority of samples have high uncertainty would be selected for annotation queries. Here a region  $\mathcal{R} \in \mathbb{R}^{nd \times nd}$  can be tiled into  $n^2$  patches. With the region-based selection, the labeled data set can be quickly expanded as

$$\mathcal{X} = \mathcal{X} \cup (\mathcal{W} \times J(\mathcal{W})), \quad (6.6)$$

where  $\mathcal{W}$  represents a set of patches from  $\mathcal{R}$ .  $J(x)$  is the assigned label for  $x$ ,  $J(\mathcal{W})$  is a set of labels  $\{J(x)\}_{x \in \mathcal{W}}$ . In the outer-loop of Fig. 6.2, we illustrate the region selection process.

## 6.4 Experiments

**Datasets and training setup.** To show the generalizability of our framework under various pathology image settings, we tested Semi-Path on two commonly-used staining techniques

(H&E and Amyloid- $\beta$ ), three tissue regions (breast, colorectal, and brain cortex), collected by three scanners (Pannoramic, NanoZoomer-XR, and Leica Aperio AT2), and three scanning magnifications (10X, 20X, and 40X). We use ResNet-18 [55] as the backbone encoder for all experiments. This framework also works with Transformer architectures in a plug-and-play setting. We select the recent SSL algorithm, FixMatch [17], as our framework’s main baseline and example. We follow the hyper-parameter settings in [17] and keep them consistent throughout all of our experiments: the initial confidence threshold  $\tau$  is set as 0.95, the unlabeled loss weight  $\lambda_u$  as 1, and the ratio of unlabeled data in each mini-batch  $\mu$  as 2. The hyper-parameters are tuned based on the validation set in the above three learning tasks. We set batch size as 32 for both labeled and unlabeled data. We use Adam [220] with  $\beta_1 = 0.9$  and  $\beta_2 = 0.999$  as our optimizer in all settings. The learning rate is set as 0.001. For comparison, all computations are conducted on a single GPU (NVIDIA Titan Xp with 12 GB of VRAM).

#### 6.4.1 Main results on pathology image classification and segmentation tasks

We first evaluate Semi-Path on two pathology classification tasks: tumor classification in breast (PCam [221]) and colorectal polyp classification (MHIST [222]). We refer to supervised ResNet-18 [55] as SL. As these are classification problems, we select Accuracy, Precision, Recall, F1-score, and AUC as the measuring metrics for the comparison. We implement recent popular SSL algorithms as baselines for comparison: Pseudo-Label [147], Mean Teacher [140], Mix-Match [30], FixMatch [17], Dash [212], and FlexMatch [18]. We also include a self-supervised learning algorithm, SimCLR [37], to compare with SSL.

**Tumor classification.** We first set the total labeling budget of PCam [221] as 1,000 images (around 0.5% of the entire training set); thus the rest of images are put into the unlabeled set. To reach the labeling budget, we set the number of IAA cycles as 2. The results are summarized in Table 6.1. We can find the performance of SL is sensitive to the amount of labeled data: the performance degrades significantly when we only use 0.5% labeled data. Then we fine-tune the self-supervised pre-trained model [37] and find the performance is still far from satisfactory due to the large gap between the new pathology task and the pre-trained domain. For SSL, the methods [17, 18, 212] that combine pseudo labeling and consistency regularization can

Table 6.1: Quantitative comparison on PCam. All scores refer to the macro-averaged values.

| Learning Manner | Labeled Ratio | Algorithm          | Accuracy                          | Precision                         | Recall                            | F1-score                          | AUC                               |
|-----------------|---------------|--------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| Semi-supervised | 100% (Full)   | ResNet-18 [55]     | $92.8 \pm 0.30$                   | $92.9 \pm 0.20$                   | $92.6 \pm 0.21$                   | $92.8 \pm 0.20$                   | $0.95 \pm 0.01$                   |
|                 | 1.5% (Full)   | ResNet-18 [55]     | $62.8 \pm 1.31$                   | $69.5 \pm 0.67$                   | $54.1 \pm 2.27$                   | $60.8 \pm 1.68$                   | $0.63 \pm 0.02$                   |
|                 | 0.5% (Full)   | ResNet-18 [55]     | $45.1 \pm 2.27$                   | $42.0 \pm 2.14$                   | $33.6 \pm 3.30$                   | $37.3 \pm 2.86$                   | $0.51 \pm 0.01$                   |
| Self-supervised | 1.5%          | SimCLR [37]        | $64.9 \pm 1.02$                   | $65.2 \pm 1.45$                   | $60.1 \pm 1.70$                   | $62.5 \pm 1.05$                   | $0.64 \pm 0.03$                   |
|                 | 0.5%          | SimCLR [37]        | $60.4 \pm 1.45$                   | $63.3 \pm 1.90$                   | $58.5 \pm 2.01$                   | $60.8 \pm 1.20$                   | $0.61 \pm 0.04$                   |
| Semi-supervised | 1.5%          | Pseudo-Label [147] | $67.9 \pm 1.85$                   | $70.5 \pm 1.90$                   | $62.2 \pm 1.76$                   | $66.1 \pm 1.42$                   | $0.67 \pm 0.04$                   |
|                 |               | Mean Teacher [140] | $68.8 \pm 1.05$                   | $69.6 \pm 1.21$                   | $62.2 \pm 1.50$                   | $65.7 \pm 1.05$                   | $0.66 \pm 0.02$                   |
|                 |               | MixMatch [30]      | $78.9 \pm 0.67$                   | $78.4 \pm 0.99$                   | $76.5 \pm 0.50$                   | $77.4 \pm 0.65$                   | $0.82 \pm 0.03$                   |
|                 |               | FixMatch [17]      | $83.9 \pm 1.29$                   | $85.1 \pm 0.85$                   | $84.1 \pm 0.95$                   | $84.6 \pm 0.90$                   | $0.87 \pm 0.01$                   |
|                 |               | Dash [212]         | $82.8 \pm 1.09$                   | $83.0 \pm 0.55$                   | $81.2 \pm 0.60$                   | $82.1 \pm 0.78$                   | $0.85 \pm 0.03$                   |
|                 |               | FlexMatch [18]     | $84.1 \pm 1.02$                   | $84.9 \pm 0.75$                   | $84.4 \pm 0.80$                   | $84.6 \pm 0.75$                   | $0.86 \pm 0.04$                   |
|                 |               | <b>Semi-Path</b>   | <b><math>89.5 \pm 1.05</math></b> | <b><math>89.9 \pm 0.70</math></b> | <b><math>89.7 \pm 1.02</math></b> | <b><math>89.8 \pm 0.78</math></b> | <b><math>0.92 \pm 0.03</math></b> |
| Semi-supervised | 0.5%          | Pseudo-Label [147] | $55.7 \pm 2.05$                   | $59.2 \pm 2.23$                   | $54.5 \pm 1.97$                   | $56.8 \pm 1.55$                   | $0.58 \pm 0.03$                   |
|                 |               | Mean Teacher [140] | $62.3 \pm 1.87$                   | $63.9 \pm 2.92$                   | $57.2 \pm 2.22$                   | $60.3 \pm 2.49$                   | $0.61 \pm 0.05$                   |
|                 |               | MixMatch [30]      | $65.6 \pm 0.34$                   | $70.2 \pm 1.20$                   | $69.0 \pm 1.45$                   | $69.6 \pm 0.95$                   | $0.75 \pm 0.06$                   |
|                 |               | FixMatch [17]      | $73.2 \pm 0.76$                   | $77.5 \pm 0.50$                   | $73.7 \pm 0.25$                   | $75.6 \pm 0.37$                   | $0.84 \pm 0.04$                   |
|                 |               | Dash [212]         | $71.0 \pm 0.98$                   | $75.3 \pm 0.78$                   | $70.2 \pm 0.45$                   | $72.7 \pm 0.50$                   | $0.82 \pm 0.02$                   |
|                 |               | FlexMatch [18]     | $74.0 \pm 0.80$                   | $78.1 \pm 1.15$                   | $73.0 \pm 0.90$                   | $75.5 \pm 0.84$                   | $0.85 \pm 0.03$                   |
|                 |               | <b>Semi-Path</b>   | <b><math>81.2 \pm 0.98</math></b> | <b><math>81.5 \pm 0.99</math></b> | <b><math>79.9 \pm 0.86</math></b> | <b><math>80.7 \pm 0.93</math></b> | <b><math>0.89 \pm 0.04</math></b> |

bring more benefits compared to SL. Semi-Path can consistently outperform the recent SSL algorithms [147, 140, 30, 17, 212, 18], with 8% improvement over FixMatch [17] and 7.2% over FlexMatch [18] in terms of accuracy. We observe similar results when we set the labeling budget as 1.5%. This shows a better utilization of the unlabeled data by Semi-Path.

**Colorectal polyp classification.** To further stress-test the applicability of Semi-Path, we select (MHIST [222]) as a more challenging learning task that suffers from inter-rater disagreement. This dataset only consists 3,152 patches but involves 328 slides. Hence the variation of feature representations from each data point in the same class can be amplified [76]. We first set 60 labeled patches (around 3% of the training set) as the total labeling budget. The results are summarized in Table 6.2. We find Semi-Path can significantly improve the performance of SSL under this challenging dataset. The performance gain is larger than 4% in all metrics. Then we increase the labeling budget to 200 images (10% of the training set) and achieve similar

Table 6.2: Quantitative comparison on MHIST. All scores refer to the macro-averaged values.

| Learning Manner | Labeled Ratio | Algorithm          | Accuracy                          | Precision                         | Recall                            | F1-score                          | AUC                               |
|-----------------|---------------|--------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| Supervised      | 100% (Full)   | ResNet-18 [55]     | 86.0 $\pm$ 0.36                   | 84.8 $\pm$ 0.61                   | 83.5 $\pm$ 0.55                   | 84.1 $\pm$ 0.58                   | 0.89 $\pm$ 0.01                   |
| Self-supervised | 10%           | SimCLR [37]        | 63.2 $\pm$ 0.95                   | 63.8 $\pm$ 1.20                   | 62.2 $\pm$ 1.70                   | 63.0 $\pm$ 0.85                   | 0.66 $\pm$ 0.04                   |
|                 | 3%            | SimCLR [37]        | 58.6 $\pm$ 0.20                   | 60.0 $\pm$ 0.55                   | 58.5 $\pm$ 0.87                   | 59.2 $\pm$ 1.20                   | 0.63 $\pm$ 0.03                   |
| Semi-supervised | 10%           | Pseudo-Label [147] | 65.4 $\pm$ 1.04                   | 66.5 $\pm$ 0.98                   | 65.9 $\pm$ 0.85                   | 66.2 $\pm$ 1.42                   | 0.75 $\pm$ 0.03                   |
|                 |               | Mean Teacher [140] | 70.1 $\pm$ 0.80                   | 71.3 $\pm$ 1.05                   | 69.9 $\pm$ 0.98                   | 70.6 $\pm$ 0.58                   | 0.80 $\pm$ 0.02                   |
|                 |               | MixMatch [30]      | 74.1 $\pm$ 0.47                   | 75.0 $\pm$ 0.75                   | 74.3 $\pm$ 0.50                   | 74.6 $\pm$ 0.97                   | 0.83 $\pm$ 0.03                   |
|                 |               | FixMatch [17]      | 70.5 $\pm$ 1.55                   | 68.8 $\pm$ 2.76                   | 69.5 $\pm$ 2.97                   | 69.1 $\pm$ 2.86                   | 0.79 $\pm$ 0.02                   |
|                 |               | Dash [212]         | 70.2 $\pm$ 1.21                   | 70.0 $\pm$ 0.85                   | 71.3 $\pm$ 0.69                   | 70.6 $\pm$ 1.48                   | 0.81 $\pm$ 0.04                   |
|                 |               | FlexMatch [18]     | 73.1 $\pm$ 0.84                   | 74.6 $\pm$ 1.05                   | 75.1 $\pm$ 0.90                   | 74.8 $\pm$ 0.89                   | 0.82 $\pm$ 0.03                   |
|                 |               | <b>Semi-Path</b>   | <b>81.1 <math>\pm</math> 0.89</b> | <b>81.0 <math>\pm</math> 0.78</b> | <b>81.2 <math>\pm</math> 0.90</b> | <b>81.1 <math>\pm</math> 0.62</b> | <b>0.87 <math>\pm</math> 0.04</b> |
|                 |               |                    |                                   |                                   |                                   |                                   |                                   |
| Semi-supervised | 3%            | Pseudo-Label [147] | 61.2 $\pm$ 1.55                   | 61.3 $\pm$ 2.00                   | 62.1 $\pm$ 1.94                   | 61.7 $\pm$ 1.67                   | 0.68 $\pm$ 0.02                   |
|                 |               | Mean Teacher [140] | 62.3 $\pm$ 1.87                   | 63.9 $\pm$ 2.92                   | 57.2 $\pm$ 2.22                   | 60.3 $\pm$ 2.49                   | 0.66 $\pm$ 0.05                   |
|                 |               | MixMatch [30]      | 66.8 $\pm$ 1.04                   | 68.2 $\pm$ 0.99                   | 67.9 $\pm$ 1.20                   | 68.7 $\pm$ 0.45                   | 0.76 $\pm$ 0.03                   |
|                 |               | FixMatch [17]      | 62.6 $\pm$ 1.35                   | 63.8 $\pm$ 1.15                   | 65.6 $\pm$ 1.35                   | 64.7 $\pm$ 1.25                   | 0.73 $\pm$ 0.03                   |
|                 |               | Dash [212]         | 64.8 $\pm$ 0.65                   | 68.3 $\pm$ 0.45                   | 67.9 $\pm$ 0.67                   | 68.1 $\pm$ 0.98                   | 0.76 $\pm$ 0.02                   |
|                 |               | FlexMatch [18]     | 63.0 $\pm$ 1.20                   | 65.8 $\pm$ 1.45                   | 64.5 $\pm$ 0.96                   | 65.1 $\pm$ 0.56                   | 0.73 $\pm$ 0.01                   |
|                 |               | <b>Semi-Path</b>   | <b>76.1 <math>\pm</math> 0.90</b> | <b>75.9 <math>\pm</math> 1.23</b> | <b>75.9 <math>\pm</math> 1.09</b> | <b>75.9 <math>\pm</math> 0.99</b> | <b>0.84 <math>\pm</math> 0.02</b> |

performance gain. These promising results show Semi-Path effectively leverages the unlabeled data under this *small-scale* setting.

**GM/WM segmentation.** Then, we evaluate Semi-Path on a gigapixel-level segmentation task. We report IoU score [38] and DICE coefficient [46] as the metrics and implement U-Net [27] as the SL baseline for segmentation problems. Semi-Path significantly outperforms FixMatch[17]: especially it almost doubles the performance in WM predictions, which is the minority class in the segmentation task. Semi-Path’s segmentation performance is also superior to SSAL [72], but the computational complexity is significantly reduced in Semi-Path compared to SSAL [72], where the framework considers the SSL and AL components as independent modules. Specifically, Semi-Path saves almost 80% of inference time in each IAA round compared to SSAL [72]. The masks can be visualized in Fig. 6.3. Hence Semi-Path is a computation-efficient framework where SSL and AL are deeply fused and share mutual information with each other.



Table 6.3: Pixel-wise IoU Scores for AD, NAD, and overall test set

| Method       | SL (FCN [11])    |                  | SL (U-Net [27])  |                         | FixMatch [17]    | ICCVW [72]       | Semi-Path        |
|--------------|------------------|------------------|------------------|-------------------------|------------------|------------------|------------------|
| Labeled data | 2 WSIs           | All WSIs         | 2 WSIs           | All WSIs                | 0.1%             | 0.1%             | 0.1%             |
| AD Back      | 61.04 $\pm$ 5.44 | 81.13 $\pm$ 9.17 | 59.74 $\pm$ 13.9 | <b>96.80</b> $\pm$ 1.48 | 93.15 $\pm$ 2.41 | 95.01 $\pm$ 1.17 | 95.09 $\pm$ 1.21 |
| AD GM        | 46.98 $\pm$ 2.78 | 76.07 $\pm$ 8.91 | 37.16 $\pm$ 9.93 | <b>89.58</b> $\pm$ 5.12 | 78.57 $\pm$ 3.87 | 88.80 $\pm$ 3.92 | 88.91 $\pm$ 4.05 |
| AD WM        | 27.75 $\pm$ 5.50 | 62.23 $\pm$ 14.0 | 7.57 $\pm$ 6.02  | <b>82.53</b> $\pm$ 7.70 | 56.66 $\pm$ 16.4 | 81.83 $\pm$ 5.53 | 81.95 $\pm$ 4.58 |
| NAD Back     | 66.66 $\pm$ 5.17 | 88.42 $\pm$ 1.55 | 78.46 $\pm$ 18.5 | <b>97.36</b> $\pm$ 3.15 | 97.07 $\pm$ 0.31 | 97.26 $\pm$ 0.52 | 97.33 $\pm$ 0.78 |
| NAD GM       | 50.15 $\pm$ 0.49 | 79.37 $\pm$ 2.95 | 59.59 $\pm$ 13.6 | <b>94.42</b> $\pm$ 3.30 | 83.97 $\pm$ 7.76 | 93.47 $\pm$ 1.60 | 93.59 $\pm$ 1.55 |
| NAD WM       | 19.72 $\pm$ 13.6 | 49.89 $\pm$ 12.8 | 3.02 $\pm$ 3.09  | <b>81.25</b> $\pm$ 9.53 | 22.72 $\pm$ 19.0 | 75.85 $\pm$ 11.4 | 77.95 $\pm$ 10.9 |
| Background   | 63.29 $\pm$ 5.81 | 84.05 $\pm$ 9.17 | 68.28 $\pm$ 17.2 | <b>97.02</b> $\pm$ 2.15 | 94.72 $\pm$ 2.71 | 95.91 $\pm$ 1.48 | 95.99 $\pm$ 1.33 |
| GM           | 48.25 $\pm$ 2.66 | 77.39 $\pm$ 7.06 | 46.13 $\pm$ 15.8 | <b>91.52</b> $\pm$ 4.94 | 80.73 $\pm$ 6.01 | 90.67 $\pm$ 3.90 | 90.78 $\pm$ 3.34 |
| WM           | 24.54 $\pm$ 9.80 | 57.29 $\pm$ 14.3 | 5.75 $\pm$ 5.37  | <b>82.02</b> $\pm$ 7.98 | 43.08 $\pm$ 24.0 | 79.44 $\pm$ 8.34 | 80.35 $\pm$ 8.67 |
| Mean         | 45.36 $\pm$ 3.26 | 72.91 $\pm$ 7.56 | 40.05 $\pm$ 10.2 | <b>90.19</b> $\pm$ 3.84 | 72.84 $\pm$ 7.18 | 88.67 $\pm$ 3.12 | 89.04 $\pm$ 2.99 |

The results are from the hold-out test set. AD refers to Alzheimer’s disease cases while NAD refers to Non-Alzheimer’s disease cases. 2 WSIs refers to 2 WSIs are labeled, equivalent to 10% regions of all WSIs; all WSIs refers to all WSIs are labeled. 0.1% refers to 0.1% regions of all WSIs are labeled, which can be tiled into 600 patches; so as 0.07% which can be tiled into 400 patches.

## 6.4.2 Empirical analysis on Semi-Path

### 6.4.2.1 Ablation Studies of Informative Active Annotation (IAA)

For this ablation study, we disable the Adaptive Pseudo Labeling (APL) module and test the module IAA on FixMatch [17]. The comparison is between the original FixMatch [17] and the combination of FixMatch [17] and IAA.

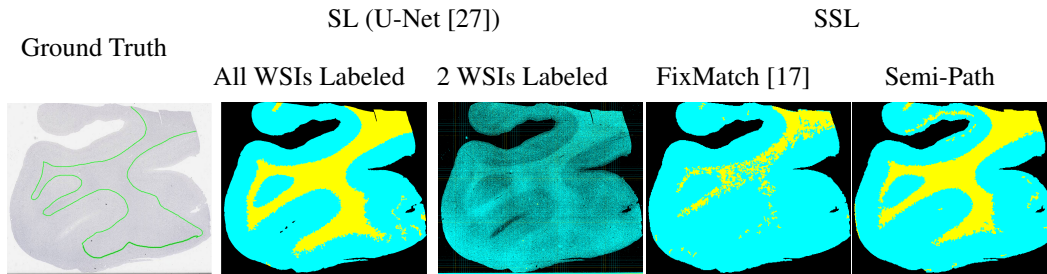


Figure 6.3. Visualization of segmentation masks: GM, WM, and background are indicated by cyan, yellow, and black, respectively.

**Effect of the rounds of IAA on eventual results.** We use a two-round IAA and show its promising improvements. In this subsection, we further study the effect of IAA rounds on eventual performance. To have a fair comparison, we set 0.5% of patches in PCam [221] as the total labeling budget. The results are summarized in Table 6.4: when we use only one-round IAA to reach the labeling budget, the improvement can be trivial; however, when we use more than four rounds to reach the budget, the performance gain is also limited. Considering the fact each AL round will inevitably incur computation costs and expert interactions, we choose two-round IAA in our study for the trade-off between the performance and clinical interactions.

Table 6.4: Effect of IAA rounds on PCam using 0.5% data with annotations. Precision, Recall, and F1-score refer to the macro-averaged values from all classes

| Rounds   | Accuracy        | Precision       | Recall          | F1-score        |
|----------|-----------------|-----------------|-----------------|-----------------|
| Original | $73.2 \pm 0.76$ | $77.5 \pm 0.50$ | $73.7 \pm 0.25$ | $75.6 \pm 0.37$ |
| 1        | $75.8 \pm 1.04$ | $79.1 \pm 1.01$ | $75.3 \pm 0.94$ | $77.2 \pm 0.99$ |
| 2        | $80.9 \pm 0.86$ | $81.0 \pm 0.84$ | $79.5 \pm 0.45$ | $80.2 \pm 0.63$ |
| 3        | $81.2 \pm 0.45$ | $81.1 \pm 0.21$ | $79.9 \pm 0.23$ | $80.5 \pm 0.34$ |
| 4        | $81.2 \pm 0.23$ | $81.2 \pm 0.12$ | $80.0 \pm 0.21$ | $80.6 \pm 0.19$ |
| 5        | $81.3 \pm 0.27$ | $81.1 \pm 0.21$ | $79.8 \pm 0.45$ | $80.4 \pm 0.37$ |

#### **Compatibility of IAA with different SSL algorithms.**

To further prove the general benefits of IAA in SSL settings, we also apply IAA to recent popular SSL algorithms to prove its general use and compatibility in SSL settings. The SSL algorithms here include Mean Teacher [140], ReMixMatch [40], and FixMatch. We summarize the results in Table 6.5. When we only set 0.5% images as the labeling budget, our proposed IAA module is able to find the informative samples for labeling and boost the performance of these original SSL algorithms consistently.

Table 6.5: Compatibility with different SSL algorithms on PCam using 0.5% data with annotations. Precision, Recall, and F1-score refer to the macro-averaged values from all classes

| SSL Algorithm      | Selection | Accuracy                          | Precision                         | Recall                            | F1-score                          |
|--------------------|-----------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| Mean Teacher [140] | Random    | $62.3 \pm 1.87$                   | $63.9 \pm 2.92$                   | $57.2 \pm 2.22$                   | $60.3 \pm 2.49$                   |
|                    | IAA       | <b><math>68.9 \pm 1.50</math></b> | <b><math>75.8 \pm 1.46</math></b> | <b><math>71.8 \pm 1.61</math></b> | <b><math>73.8 \pm 1.54</math></b> |
| ReMixMatch [40]    | Random    | $70.1 \pm 2.69$                   | $73.7 \pm 2.13$                   | $70.7 \pm 2.03$                   | $72.2 \pm 2.05$                   |
|                    | IAA       | <b><math>78.7 \pm 0.95</math></b> | <b><math>79.2 \pm 0.66</math></b> | <b><math>78.4 \pm 0.44</math></b> | <b><math>78.8 \pm 0.55</math></b> |
| FixMatch [17]      | Random    | $73.2 \pm 0.76$                   | $77.5 \pm 0.50$                   | $73.7 \pm 0.25$                   | $75.6 \pm 0.37$                   |
|                    | IAA       | <b><math>80.9 \pm 0.86</math></b> | <b><math>81.0 \pm 0.84</math></b> | <b><math>79.5 \pm 0.45</math></b> | <b><math>80.2 \pm 0.63</math></b> |

#### 6.4.2.2 Computational efficiency of IAA in Semi-Path

Previous AL systems involve an inference stage where the trained model will traverse all unlabeled data and apply the acquisition function for ranking [188]. The computational complexity can be heavy even if we use the simplistic acquisition functions given that the WSIs in pathology are at gigapixel level, which is much larger than other natural images. To reduce computation cost, the IAA module in Semi-Path has a two-loop selection process: fuse SSL and AL so that the training information from SSL training can be used to construct a much smaller but more effective candidate pool for labeling. Hence Semi-Path can avoid making inferences over the entire unlabeled data but focus on a much smaller pool. We save almost 80%, 70%, and 80% of inference time in PCam [221], MHIST [222], and GM/WM dataset [72], respectively. In GM/WM dataset, the original inference time for WSIs can be nearly 20.7 hours [72] on a single GPU (Nvidia Titan Xp), whereas Semi-Path only needs 3.8 hours (as shown in Table 6.6), which saves over 80% of computation time compared to [72]. Therefore, we believe this is one of the advantages of IAA in Semi-Path because its inner-outer selection structure can effectively avoid unnecessary inference on those confident samples.

#### 6.4.2.3 *Max-Entropy* vs. *Variation-Ratios*

We have provided the proof on the equivalence of *Max-Entropy* [218] and *Variation-Ratios* [217] under binary classification problems. We report their performance comparison in the GM/WM

segmentation dataset with three AL cycles. As shown in Table 6.6: both acquisition criteria used in IAA achieve similar performance and significantly reduce the AL inference time.

Table 6.6: Performance comparison between two uncertainty-based criteria on GM/WM task [72] with 0.1% labeled regions by FixMatch [17] and three-round AL

| Methods                       | IoU score |       | DICE  |       | AL Inference Time |
|-------------------------------|-----------|-------|-------|-------|-------------------|
|                               | GM        | WM    | GM    | WM    |                   |
| FixMatch [17]                 | 80.73     | 43.08 | 91.02 | 63.19 | -                 |
| ICCVW [72]                    | 90.67     | 79.44 | 95.19 | 88.53 | 20.7 hours        |
| <i>Max-Entropy</i> [218]      | 90.81     | 80.41 | 96.12 | 89.94 | 4.90 hours        |
| <i>Variation-Ratios</i> [217] | 90.78     | 80.35 | 96.04 | 90.01 | 3.80 hours        |

#### 6.4.2.4 Data imbalance scenario

To explore the source of performance gain, we take a deeper look into per class’s performance. As we achieve more than 10% of improvement on MHIST [222], we report the performance from each IAA round to visualize the benefits for each class. MHIST [222] is a challenging dataset due to its limited amount and data imbalance issue [76]. As shown in Figure 6.4, our framework improves the performance (in terms of Recall and Precision) on sessile serrated adenomas (SSA), which demonstrates its effectiveness in classifying the minor class in this dataset. Recent theoretical works [32, 41] found the generated pseudo labels from minor classes during SSL training can be significantly underestimated, which subsequently biases the model towards the significant classes and degrades the performance. In other words, more minority samples have never been assigned pseudo labels during the training process. These samples will be split into the candidate pool in our two-loop structure of IAA. Subsequently, query for annotations and enter the labeled set. Therefore, our framework has the potential to realign the underlying data imbalance in the unlabeled pool and boost the applicability of SSL in real-world applications.

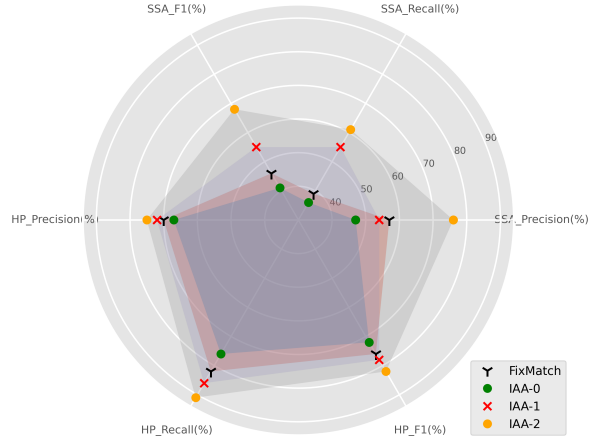


Figure 6.4. Comparison on per class’s performance in MHIST [222] under the labeling budget as only 60 labeled images. IAA-0: the starting stage (20 labeled images ), IAA-1: Semi-Path after 1 IAA round (40 labeled data); IAA-2: Semi-Path after 2 IAA rounds (60 labeled images).

## 6.5 Discussion

We propose Semi-Path, a deployable and active semi-supervised learning framework for relieving labor efforts for gigapixel-resolution pathology applications. We first design class-aware Adaptive Pseudo Labeling (APL) with dynamic thresholds to select the unlabeled samples for pseudo-labeling based on the learning status of SSL and class difficulty. Then we design and incorporate Informative Active Annotation (IAA) in our SSL framework to further exploit the informativeness in the unlabeled pool. We evaluate Semi-Path on three pathology datasets and show its consistent improvement over other SSL algorithms. Additionally, Semi-Path reduces computational complexity for the AL rounds since SSL and AL are integrated into a unified framework with IAA. We believe Semi-Path has the potential to enhance the applicability of SSL in pathology image applications. The limitation of Semi-Path is that we assume the unlabeled pool has the same classes as the labeled set.

# Chapter 7

## **PADCLIP: Pseudo-labeling with Adaptive Debiasing in CLIP for Unsupervised Domain Adaptation**

Traditional Unsupervised Domain Adaptation (UDA) leverages the labeled source domain to tackle the learning tasks on the unlabeled target domain. It can be more challenging when a large domain gap exists between the source and the target domain. A more practical setting is to utilize a large-scale pre-trained model to fill the domain gap. For example, CLIP shows promising zero-shot generalizability to bridge the gap. However, after applying traditional fine-tuning to specifically adjust CLIP on a target domain, CLIP suffers from catastrophic forgetting issues where the new domain knowledge can quickly override CLIP’s pre-trained knowledge and decrease the accuracy by half. We propose Catastrophic Forgetting Measurement (CFM) to adjust the learning rate to avoid excessive training (thus mitigating the catastrophic forgetting issue). We then utilize CLIP’s zero-shot prediction to formulate a Pseudo-labeling setting with Adaptive Debiasing in CLIP (PADCLIP) by adjusting causal inference with our momentum and CFM. Our PADCLIP allows end-to-end training on source and target domains without extra overhead. We achieved the best results on four public datasets, with a significant improvement (+18.5% accuracy) on DomainNet.

## 7.1 Introduction

Unsupervised Domain Adaptation (UDA) proposes to reduce data annotation costs by leveraging a labeled source domain to transfer the knowledge into an unlabeled target domain [47, 48, 49, 50, 51]. Prior UDA works focus on bridging the domain gap between source and target domains [51, 52, 53, 54], or increasing network capacity [49, 48] by changing a convolutional neural network (e.g., ResNet [55]) to Vision Transformer (ViT) [56]. All of these past methods are pre-trained on ImageNet [57], but large-scale pre-training is becoming practical and achieves superior performance in many fields [58, 59, 60, 61]. In theory, if the pre-trained dataset is large enough, the domain gap between source and target domains could be bridged by the pre-trained dataset itself. Hence, we argue that large-scale pre-training is an important missing part of UDA.

We choose CLIP [59], a vision-language model pre-trained on 400 million image-text pairs. Without fine-tuning, CLIP outperforms SSRT [48], a state-of-the-art UDA method on DomainNet [66]. This is thanks to the large-scale training set, which allows CLIP to disentangle object class from object domain (e.g., “a photo of a dog” vs “a sketch of a dog”): the language supervision in the form of a sentence used by CLIP is more descriptive than a single class label. However, on VisDA-2017, CLIP without fine-tuning underperforms previous work, SDAT [223]. This is because the synthetic data generated from the real-world domain do not exist in CLIP’s training set, so we still need to fine-tune CLIP to adapt it for a specific domain task.

We first adopt the traditional approach to fine-tune CLIP on VisDA-2017 [67] but found that CLIP suffers from catastrophic forgetting issues. As shown in Fig.7.1: before fine-tuning, CLIP has a strong representation power that can achieve 67% top-1 ImageNet accuracy, while dropping to 22% after fine-tuning on VisDA-2017. The loss of CLIP’s representation power causes the accuracy degradation on VisDA-2017. To counter this, it is possible to preserve CLIP’s representation power by fine-tuning both CLIP and VisDA-2017 datasets jointly, but CLIP requires several weeks to train a single setting (DomainNet has 30 settings, so it takes a year for a single experiment). Moreover, we anticipate the data imbalance issue during joint training since CLIP’s training set is 142 times larger than VisDA-2017. We seek a more practical solution for catastrophic forgetting issues without adding extra overheads.

We attempt to fine-tune CLIP on the UDA dataset with a lower learning rate and observe

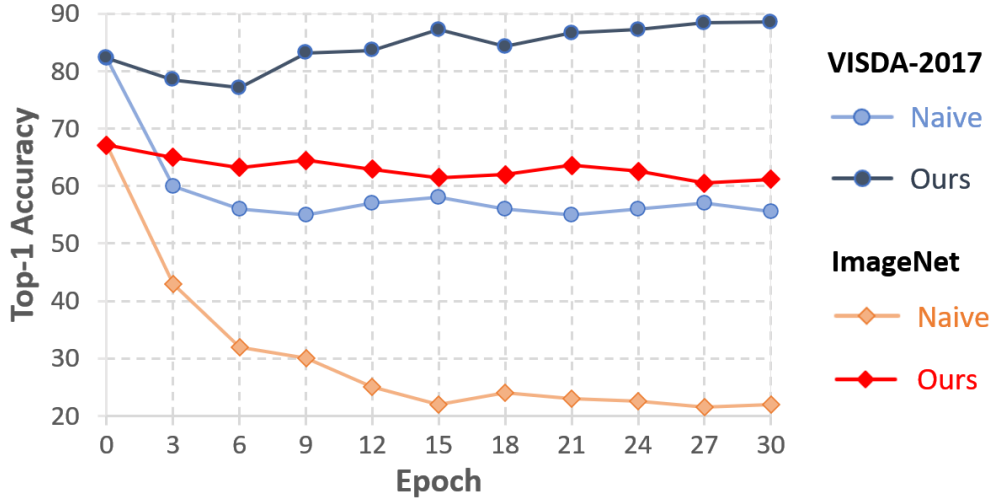


Figure 7.1. Catastrophic Forgetting. We naively fine-tune CLIP (ResNet-101) on VisDA-2017 source domain training set, and test it on validation sets of VisDA-2017 target domain and ImageNet-1K. CLIP forgets pre-trained knowledge (ImageNet accuracy -45%), resulting in -27% VisDA-2017 accuracy. Our PADCLIP mitigates catastrophic forgetting issues to achieve +6% VisDA-2017 accuracy.

less catastrophic forgetting issues, but the low learning rate prevents CLIP from learning new knowledge. To solve this problem, we propose to adjust the learning rate with Catastrophic Forgetting Measurement (CFM, Fig. 7.2) by comparing the original representation (forward original image on original CLIP) and fine-tuned representations (forward augmented images on fine-tuned CLIP). CFM is, however, unstable because every image has a different forgetting rate, so we leverage our observation that CLIP is likely to have similar predictions across all augmentations when the training example is easy (and large difference for the hard example). We propose to measure the consistency between weak (translate, flip) and strong augmentation (perturb visual appearance) as a momentum (Dual Consistency Momentum, DCM) to stabilize CFM. Our method does not introduce extra overhead: since the augmentation is already a part of fine-tuning, original prediction can be cached, and we do not need to fine-tune UDA and CLIP datasets jointly.

We further seek to use CLIP with pseudo-labeling on the target domain, which recently enjoyed success in UDA [62, 63, 49, 64, 63]. DebiasPL [65] utilized CLIP for pseudo-labeling, but it was designed for a single domain. After extending to source and target domains (UDA setting), DebiasPL[65] suffers from catastrophic forgetting issues (accuracy decreases by 21%



on VisDA-2017 after fine-tuning). To solve this problem, we replace the fixed debias factor in DebiasPL[65] with our CFM, and replace the fixed momentum in DebiasPL with our adaptive momentum (DCM). We further include a domain name into a prompt (such as: “This is a [**sketch**] photo of [car]”). Our method mitigates the catastrophic forgetting issue, and achieves the best results on DomainNet [66], VisDA-2017 [67], Office-Home [68], Office-31 [69]. To summarize, our main contributions are:

- We propose to use CLIP in UDA and discover the catastrophic forgetting issue when fine-tuning CLIP. We propose CFM for CLIP in UDA to mitigate this issue without introducing extra computational overhead.
- We propose pseudo-labeling for CLIP in UDA by extending DebiasPL to multiple domains, and replacing debias factor and momentum with our CFM and DCM. We also introduce a domain name into a prompt.
- We achieve the best results on four benchmarks on both ResNet and ViT, with a large performance improvement on the large-scale dataset (+18.5% accuracy on DomainNet).

## 7.2 Related works

**Unsupervised Domain Adaptation (UDA)** adapts a model trained on a source domain with annotated datasets to an unlabeled target domain. Discrepancy-based methods [224, 225, 226, 227] learn domain-invariant features via minimizing the discrepancy between source and target domains, or applying adversarial learning [224, 225, 226, 228, 223, 229] to obtain domain-invariant representations. DANN [47] and CDAN [228] use a domain discriminator to classify source and target samples while the feature extractor tries to fool the domain discriminator with generated domain-invariant features. AaD [230] proposes prediction consistency on the neighboring features, while CPR [231] uses prototypes instead. In theory, our improvement is orthogonal to these methods, and they could complement each other. Recent works found that the cross-attention in Vision Transformer (ViT) [56] is advantageous to feature alignment, and it is more robust to the noisy samples compared to CNN, which is the key to the UDA task [49]. Hence recent works [49, 48, 223, 232] use ViT as the backbone and achieve better performance than CNNs. We also leverage ViT.

**Pseudo-labeling in UDA** aims to leverage unlabeled target domain data [62, 63, 49, 48, 233, 234]. Specifically, the model will generate pseudo labels on the unlabeled data during the training process and use them as the supervision in the following training loop. Consistency regularization is also applied to different disturbed views of the same samples [17, 18, 235] to promote the prediction consistency on the unlabeled data. However, these methods are built on the assumption that the unlabeled data share similar distributions as the labeled data [41, 65, 233, 236], which is usually not true since the source and target are in different domains. Such distribution mismatch may generate low-quality and biased pseudo labels [41], resulting in a poor-performance classifier during the self-training process. Therefore, to relieve the noise and bias, we focus on generating less-biased pseudo labels during the UDA training in this work.

**Vision-Language Models** have shown promising results in learning generic visual representations [59, 237, 238, 239]. Recent models gain their advancement via text representation learning with Transformers [240], contrastive representation learning, and web-scale training datasets [241]. For example, CLIP [59] was trained on 400 million image-text pairs and achieved state-of-the-art performance in many fields [58, 59, 60, 61]. However, the best way to adapt CLIP for downstream tasks is still under study. For example, DebiasPL [65] found that CLIP [59] produces imbalanced prediction and proposes to adaptively debias pseudo-label for a single domain. We extend DebiasPL [65] from a single domain to multiple domains in the UDA setting, and address the domain gap problem by mitigating catastrophic forgetting issues, introducing domain name into the prompt, and dynamically adjusting debias strength and momentum.

**Causal Inference** has been introduced in computer vision tasks to alleviate the dataset bias in domain-specific applications [242, 243, 244, 245, 246, 247, 248]. These methods successfully improve performance in many fields such as image classification [249, 250], semantic segmentation [251], visual representation learning [252] and image captioning [253]. Counterfactual inference is a popular method that was used to capture the bias as the direct causal effect [254], eliminate the confounding effect [243], and disentangle the desired direct effect [255]. Our method is built on top of these works to debias pseudo-labels in our setting.

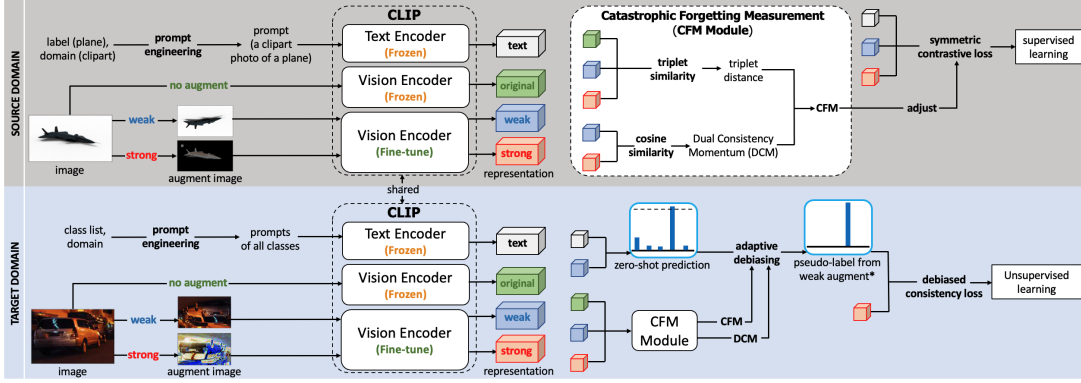


Figure 7.2. Overview. For the source domain, we convert the label and domain name into a prompt, and obtain text and image representations to train CLIP in a supervised manner. We use CLIP’s original representation, and weak/strong augmented representations to measure CFM to adjust the learning rate for mitigating catastrophic forgetting issues. For the target domain, we use zero-shot prediction in CLIP to obtain pseudo-labels and adaptively debias them with CFM (adjust debias factor) and DCM (adjust momentum) for unsupervised learning. \*Pseudo-label is converted into a **prompt** to obtain text representation.

## 7.3 Methodology

Given a labeled source domain  $\mathcal{D}_s = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{N_s}$  and an unlabeled target domain  $\mathcal{D}_t = \{(\mathbf{x}_i^t)\}_{i=1}^{N_t}$ , we aim to optimize a model from the labeled source domain to the unlabeled target domain.  $N_s$  and  $N_t$  denote the size of the source and target datasets respectively.

### 7.3.1 CLIP in UDA

We first modify CLIP to be suitable for UDA tasks. CLIP [59] is composed of a vision encoder  $f$  (maps image into low dimensional image representations), and a text encoder  $g$  (maps sentence into text representations). CLIP requires image-text pairs to jointly train  $f$  and  $g$  with symmetric cross-entropy loss [256] between the image and text representations. We follow prompt engineering [59] to prepare image-text pairs in UDA datasets. Our label  $y_z^s$  denotes a sentence in the format of “a [DOMAIN] photo of a [CLASS]”, where [CLASS] is a classification class name and [DOMAIN] is a domain name in UDA tasks (e.g., a synthetic photo of a person). During testing, we follow CLIP zero-shot inference by comparing image representations with the classification weights generated by the text encoder, denoted as  $\{\theta_z\}_{z=1}^K$ . By forwarding  $K$  descriptions corresponding to  $K$  classes, we can calculate the probability that a training image

Table 7.1: Baseline. We fine-tune CLIP (ResNet-101) on VisDA-2017 source domain training set, and test on VisDA-2017 target domain validation set. “No fine-tuning” is the best baseline because catastrophic forgetting issues are so severe such that any fine-tuning will cause accuracy drops.

| # | Configuration                  | Accuracy |
|---|--------------------------------|----------|
| 1 | No fine-tuning                 | 82.3%    |
| 2 | Fine-tune 30 epoch             | 55.6%    |
| 3 | Early stop (fine-tune 1 epoch) | 73.2%    |
| 4 | Lower learning rate 50x on #2  | 75.8%    |

belongs to the  $k$ -th category.

$$\ddot{p}_k = P(\hat{y}_z^t = k | \mathbf{x}^t) = \frac{\exp(\cos(\boldsymbol{\theta}_k, f(\mathbf{x}^t))/T)}{\sum_{z=1}^K \exp(\cos(\boldsymbol{\theta}_z, f(\mathbf{x}^t))/T)} \quad (7.1)$$

where  $T$  is the temperature parameter learned by CLIP,  $\cos$  refers to cosine similarity [59], and we denote a vector of  $\ddot{p}_k$  as  $p$  (probability of a sample in a minibatch).

### 7.3.2 Catastrophic Forgetting Measurement

After formulating CLIP for UDA and attempting to fine-tune CLIP on UDA datasets, we found that CLIP suffers from catastrophic forgetting issues (Fig. 7.1). We explored baselines in Tab. 7.1 and found that the original CLIP (no fine-tuning) performs the best (any fine-tuning will cause performance degradation), but the degradation can be mitigated by lowering the learning rate. We propose to decrease the learning rate according to the difference between the original CLIP ( $o$ ) and fine-tuned CLIP ( $f$ )’s representations. Large differences indicate that CLIP forgets the pre-trained knowledge (resulting in a new representation). The original CLIP’s representation can be cached by running CLIP on the original image (no augmentation) before fine-tuning, and we seek a meaningful augmentation during fine-tuning.

We follow CLSA [257] to introduce “weak” (no appearance change: translate, flip) and “strong” augmentation (perturb appearance: CTAugment [40], RandAugment [202]), and propose to measure the distance between representations from both augmentations. We lower the learning rate when the difference is large because we observe more mistakes from “strong” ( $s(x^s)$ ) than “weak” ( $w(x^s)$ ) augmentation predictions when CLIP struggles with the hard training example

(leads to more forgetting). Combining both of our proposals, we formulate a triplet of original CLIP’s representations from the original image ( $o(x^s)$ ), fine-tuned CLIP’s representations for “weak” ( $f(w(x^s))$ ) and “strong” ( $f(s(x^s))$ ) augmentation. We use Euclidean Distance to measure the similarity of each pair, and we sum all pairs into our triplet distance ( $\lambda^s$ ). We flip the sign to lower the learning rate when the difference of each pair is large.

$$\lambda^s = 1 - \frac{1}{6B} \sum_{i=1}^B (\|\tilde{f}(w(x_i^s)) - \tilde{o}(x_i^s)\|_2 + \|\tilde{f}(s(x_i^s)) - \tilde{o}(x_i^s)\|_2 + \|\tilde{f}(w(x_i^s)) - \tilde{f}(s(x_i^s))\|_2) \quad (7.2)$$

where  $\sim$  denotes representation normalization with L2 norm (e.g.,  $\tilde{o}(x^s) = o(x^s)/\|o(x^s)\|_2$ ) to cap each representation between  $[-1, 1]$ . The summation of all terms is in the range of  $[0, 6]$ , so we divide the summation by 6 to cap  $\lambda^s$  range to  $[0, 1]$ , and average distances over the batch size ( $B$ ).  $\lambda^s$  is, however, unstable because the input ( $x^s$ ) changes every iteration, so the learning rate will constantly be adjusted (resulting in unstable training). To solve this problem, we propose a momentum to slow down the change of  $\lambda^s$ . We measure the consistency between “weak” and “strong” representations to use as Dual Consistency Momentum (DCM,  $m^s$ ).

$$m^s = \frac{1}{B} \sum_{i=1}^B \cos(f(w(x_i^s)), f(s(x_i^s))) \quad (7.3)$$

where  $\cos$  is a cosine similarity. We use low momentum (small  $m^s$ , slow  $\lambda^s$  changes) when the consistency is low (indicating a hard training example). We combine triplet distance with DCM to define Catastrophic Forgetting Measurement (CFM,  $\lambda_z^s$ ) for each iteration ( $z$ ) with  $\lambda_0^s = 0$ .

$$\lambda_z^s \leftarrow m^s \lambda_z^s + (1 - m^s) \lambda_{z-1}^s \quad (7.4)$$

### 7.3.3 Pseudo-labeling and inter-class bias

**Pseudo-labeling in CLIP.** Pseudo-labeling [17, 18] enjoyed success in UDA by leveraging the unlabeled target domain data, but past methods [62, 63, 49, 48] were not designed for CLIP. DebiasPL [65] supports pseudo-labeling for CLIP, but was designed for a single domain. We first follow DebiasPL to generate a soft label on “weak” augmented samples from the target domain as  $q = p(y|w(\mathbf{x}^t))$ , convert a soft label into a hard label by a one-hot encoder ( $\mathbb{1}$ ), and use a

Table 7.2: Our improvements on DebiasPL. We fine-tune DebiasPL (ResNet-101) on VisDA-2017 source domain training set, and test on VisDA-2017 target domain validation set. Our improvement mitigates catastrophic forgetting issues, and improve pseudo-label quality.

| # | Configuration                                                      | Accuracy |
|---|--------------------------------------------------------------------|----------|
| 1 | DebiasPL [65]                                                      | 64.4%    |
| 2 | Add CFM to supervised loss ( $\mathcal{L}_{sup}$ , Eq. 7.12) on #1 | 85.9%    |
| 3 | Add domain name to prompt on #2                                    | 86.5%    |
| 4 | Add CFM to pseudo-label (Eq. 7.10) on #3                           | 87.8%    |
| 5 | Add DCM to pseudo-label (Eq. 7.9) on #4                            | 88.5%    |

fixed threshold ( $\tau = 0.4$ ) to select high confident pseudo-labels. We formulate a consistency loss ( $\mathcal{L}_{cp}$ ) by using a cross-entropy loss ( $\mathbf{H}$ ) to push the prediction from “strong” augmentation ( $p(y|s(\mathbf{x}^t))$ ) to be closed to pseudo-label from “weak” augmentation.

$$\mathcal{L}_{cp} = \frac{1}{B} \sum_{i=1}^B \mathbb{I}[\max(q_i) \geq \tau] \cdot \mathbf{H}(p(y_i|s(\mathbf{x}_i^t)), q_i) \quad (7.5)$$

**Inter-class Bias in Pseudo-labeling** is caused by the reliance on a trained model to generate pseudo-labels. If multiple classes have similar appearances (e.g., dog vs. wolf), the model tends to have prediction errors, which will generate incorrect pseudo-labels. As the training goes on, these incorrect pseudo-labels will further increase the existing bias and eventually lead to a significant accuracy drop. Past pseudo-labeling debiased methods [65, 41] perform reasonably well in a generic setting, but the bias in the UDA setting is more severe due to the domain gap between source and target domains. We analyze the pseudo-label quality in Fig. 7.3 and observe the co-existence effect [258] where many samples in the confusing classes tend to be misclassified into other similar classes (e.g., “bus”, “car”, and “truck” are all belong to “vehicle”). We seek to simultaneously mitigate inter-class bias while combating catastrophic forgetting.

### 7.3.4 Pseudo-labeling with Adaptive Debiasing

**Causal Inference.** We follow DebiasPL [65] to use causal inference to mitigate inter-class bias in pseudo-labels. Given the causal graph in Fig. 7.4, debiasing of predictions can be delineated as

the direct causal effect along  $x_i^t \rightarrow p$ , defined as Controlled Direct Effect (CDE) [259, 258, 65].

$$\text{CDE}(p_i) = [p_i | do(x_i^t), do(D)] - [p_i | do(\hat{x}^t), do(D)] \quad (7.6)$$

where  $do(\cdot)$  denotes the causal intervention [260] that removes the model bias ( $M$ ) from  $x^t$ , and  $\hat{x}^t = \{x_1^t, \dots, x_n^t\}$ . It is, however, computationally expensive to visit all training samples to measure the counterfactual outcome.

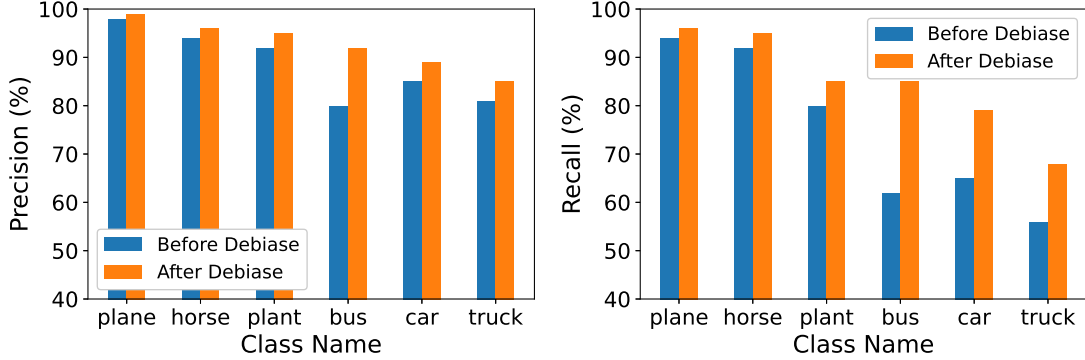


Figure 7.3. Debaised Pseudo-labeling. Inter-class bias degrades pseudo-label quality and reduces the recall of classes with similar appearance (e.g., bus, car, truck). Our debaised method can mitigate such issues and improve recall.

**Debiasing by DebiasPL [65].** We follow DebiasPL [65] to use Approximated Controlled Direct Effect (ACDE) by assuming the model bias is not drastically changed. This assumption holds true in our settings as we aim to fine-tune CLIP and keep the original knowledge. We approximate the first term ( $[p_i | do(x_i^t), do(M)]$ ) in Eq. 7.6 as

$$p'_i \leftarrow \ddot{m} p'_i + (1 - \ddot{m}) \frac{1}{B} \sum_{i=1}^B p_i \quad (7.7)$$

where  $p_i$  is the vector of prediction from Eq. 7.1,  $p'$  is the debaised prediction, and  $\ddot{m}$  is a fixed momentum. Then the debaised pseudo-label can be formulated as

$$q'_i = q_i - \mu \log p'_i \quad (7.8)$$

where  $\mu$  is a fixed debias factor,  $q$  is a soft label from Eq. 7.5, and  $q'$  is the debaised soft label.

**Debiasing by CFM.** DebiasPL [65] was designed for a single domain (DebiasPL’s setting splits a single dataset (such as CIFAR-10) into labeled (source) and unlabelled (target) data). Hence,

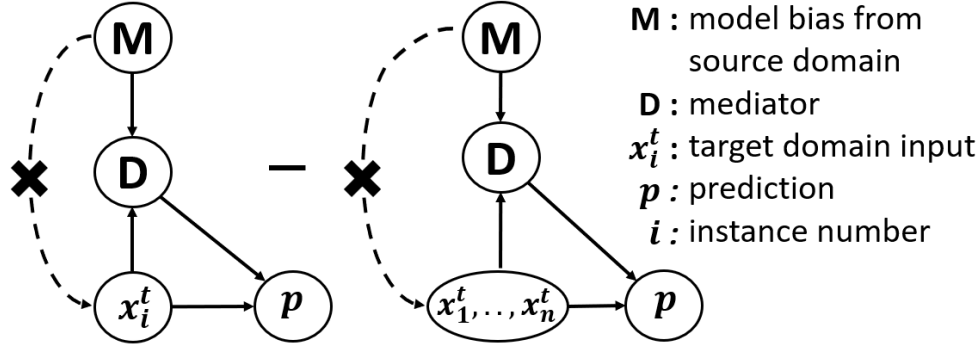


Figure 7.4. Causal graph via counterfactual reasoning for debiasing model predictions for pseudo-labels.

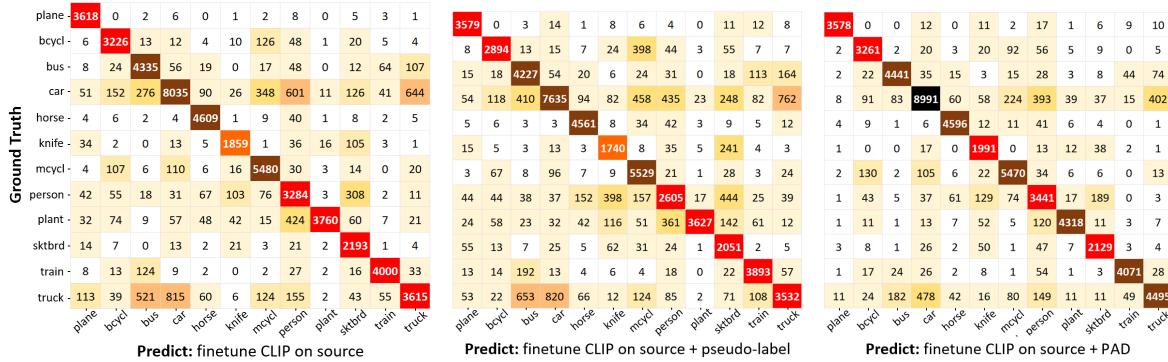


Figure 7.5. The confusion matrix on VisDA-2017 shows the effect of pseudo-label and the pseudo-label with adaptive debiasing (PAD). Diagonal values are true positive (darker = better) and other values are errors (brighter = better).

it suffers from severe bias due to the domain gap in UDA (Tab. 7.2). We identify the debias factor ( $\mu$ , Eq. 7.8) as one of the root cause because  $\mu$  is sensitive (small  $\mu$  does not eliminate bias, and large  $\mu$  prevents the model from learning new knowledge [65, 258]). Moreover,  $\mu$  is set to a fixed value, but the bias is dependent on the domain setting (e.g., real-world vs. synthetic will likely have a higher bias than sketch vs. quick draw), so we propose to adaptively adjust  $\mu$ . We incorporate catastrophic forgetting information by replacing  $\mu$  with CFM ( $\lambda_z^t$ ) to adjust  $\mu$  adaptively, and replace the fix momentum ( $\dot{m}$ , Eq. 7.7) with DCM ( $m^t$ ) to also adjust  $\dot{m}$  adaptively. Both CFM and DCM for debiasing are computed on target domain input ( $x^t$ ).

$$p'_i \leftarrow m^t p'_i + (1 - m^t) \frac{1}{B} \sum_{i=1}^B p_i \quad (7.9)$$

$$q'_i = q_i - \lambda_z^t \log p'_i \quad (7.10)$$

Finally, the debiased consistency loss ( $\mathcal{L}_{dcp}$ ) is formulated by replacing  $q$  with  $q'$  on the consis-



tency loss (Eq. 7.5).

$$\mathcal{L}_{dcp} = \frac{1}{B} \sum_{i=1}^B \mathbb{1}[\max(q'_i) \geq \tau] \cdot \mathbf{H}(p(y_i|s(\mathbf{x}_i^t)), q'_i) \quad (7.11)$$

### 7.3.5 End-to-end trainable pipeline

Fig. 7.2 shows the pipeline. We use symmetric contrastive loss [59] ( $\mathcal{L}_{sup}$ ) and debiased consistency loss ( $\mathcal{L}_{dcp}$ ).

$$\mathcal{L} = \lambda_z^s \mathcal{L}_{sup}(\mathcal{D}_s) + \Lambda \mathcal{L}_{dcp}(\mathcal{D}_t) \quad (7.12)$$

where  $\Lambda = 0.5$  is a constant term for adjusting  $\mathcal{L}_{dcp}$ . We use CFM ( $\lambda_z^t$ , Eq. 7.2) to adjust  $\mathcal{L}_{sup}$  to mitigate catastrophic forgetting issues. Our loss formulation allows end-to-end training to simultaneously preserve CLIP knowledge, supervised fine-tuning on the source domain, and fine-tuning with debias pseudo-labels on the target domain.

## 7.4 Experimental setup

**Dataset.** We evaluate our proposed methods on four popular UDA datasets. **VisDA-2017** [67] contains 152k synthetic images and 55k real object images of 12 categories sampled from Microsoft COCO. **Office-Home** [68] has 15.5k images of 65 categories from 4 domains: Art (Ar), Clipart (Cl), Product (Pr), and Real-world (Rw). **Office-31** [69] includes 31 classes and 4.6k images from 3 domains: Amazon (A), DSLR (D), and Webcam (W). **DomainNet** [66] is the most challenging dataset that contains 0.6 million images of 345 classes from 6 domains: Clipart (clp), Infograph (inf), Painting (pnt), Quickdraw (qdr), Real-world (rel), and Sketch (skt). We strictly follow the protocol of previous works [48, 49, 223]. On VisDA-2017, we use synthetic as the source domain and the real object as the target domain; on the other three datasets, we select one domain as the source and another domain as the target. We have 12, 6, and 30 source-target combinations on Office-Home, Office-31, and DomainNet, respectively.

**Training Configuration.** We experiment on both ViT-B [56] (patch size  $16 \times 16$ , batch size 16) and ResNet [55] (batch size 32) as the vision encoder in CLIP [59]. The learning rate is set to  $1e^{-6}$  on all datasets, except  $1e^{-7}$  on VisDA-2017 because training on VisDA-2017’s synthetic data is not stable and the training may diverge. We freeze the text encoder and only train the vision encoder in the CLIP framework. We follow the training process in CLIP to use Adam

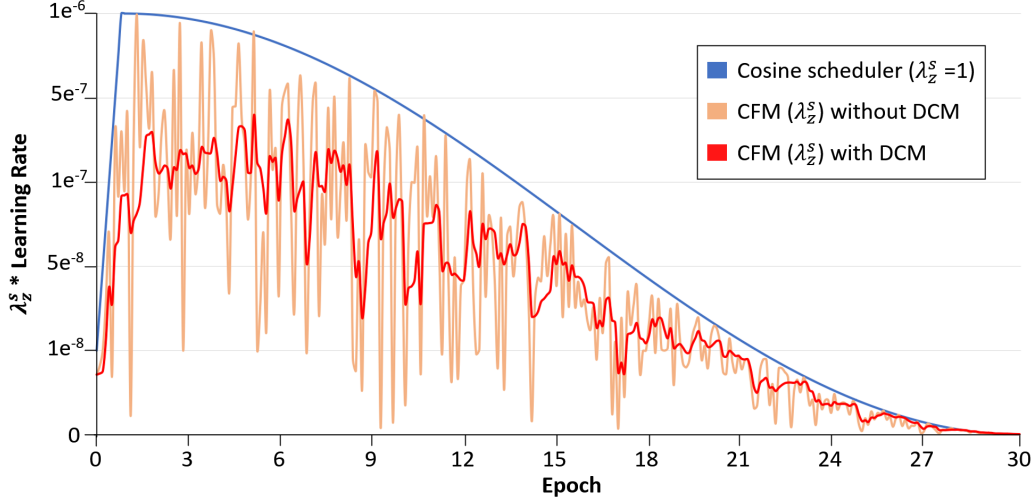


Figure 7.6. We use cosine scheduler with base learning rate (base  $LR$ )  $1e^{-6}$  (except VISDA 2017 ( $1e^{-7}$ )) and warm up for 1 epoch. CFM ( $\lambda_z^s$ ) is the loss weight for the source domain ( $L_{sup}$ ) and CFM will adjust the gradient together with the learning rate ( $\lambda_z^s * LR$ ). Low CFM (high catastrophic forgetting) will decrease the gradient, and we further smooth CFM with momentum (DCM).

optimizer with decoupled weight decay regularization [261] incorporated into all weights that are not gains or biases. Cosine schedule [262] is used to decay the learning rate and we train every setting for 30 epochs.

## 7.5 Results

### 7.5.1 Ablation studies

**Importance of Pretrained Data.** Since CLIP is data-hungry [59, 237, 238, 239], we first study the sensitivity of CLIP in the UDA setting. We compare public models from CLIP [59] (trained on 400 million image-text pairs) with OpenCLIP [263] (trained on Conceptual Captions dataset [264], 3 million image-text pairs). Tab. 7.3 shows the accuracy drops from 82% to 59% when the pre-trained dataset is reduced. Therefore, although CLIP has a strong zero-shot generalization, such capability is learned through large-scale data. In contrast, a small dataset is likely to lack enough samples to disentangle the domain and class. We conclude that a large pre-trained dataset is important, but we keep both datasets throughout our ablation studies to show that our improvement holds true even with small pre-trained data.

Table 7.3: Ablation study. We fine-tune CLIP (ResNet-101) on VisDA-2017 source domain training set, and test on VisDA-2017 target domain validation set. CLIP, trained on 400M, outperforms 3M (row: 1,6). Finetune CLIP on a source domain leads to catastrophic forgetting and CFM can mitigate it (row: 6-8). Pseudo-label (PL) with Adaptive Debiasing (PAD) further improves accuracy (row: 8-10).

| #  | Pretrain | CLIP | Source | CFM | PL | PAD | Accuracy |
|----|----------|------|--------|-----|----|-----|----------|
| 1  | 3 M      | ✓    | ✗      | ✗   | ✗  | ✗   | 59.1%    |
| 2  | 3 M      | ✓    | ✓      | ✗   | ✗  | ✗   | 44.3%    |
| 3  | 3 M      | ✓    | ✓      | ✓   | ✗  | ✗   | 62.9%    |
| 4  | 3 M      | ✓    | ✓      | ✓   | ✓  | ✗   | 65.7%    |
| 5  | 3 M      | ✓    | ✓      | ✓   | ✓  | ✓   | 67.1%    |
| 6  | 400 M    | ✓    | ✗      | ✗   | ✗  | ✗   | 82.3%    |
| 7  | 400 M    | ✓    | ✓      | ✗   | ✗  | ✗   | 55.6%    |
| 8  | 400 M    | ✓    | ✓      | ✓   | ✗  | ✗   | 83.9%    |
| 9  | 400 M    | ✓    | ✓      | ✓   | ✓  | ✗   | 86.0%    |
| 10 | 400 M    | ✓    | ✓      | ✓   | ✓  | ✓   | 88.5%    |

**Importance of CFM.** CLIP suffers from catastrophic forgetting issues (Fig. 7.1), so we use Catastrophic Forgetting Measurement (CFM) to compare representations from the fine-tuned CLIP against the original CLIP. CFM adjusts the learning rate to slow down the forgetting process while accumulating new knowledge from the UDA dataset (Fig. 7.6). Tab. 7.3 shows that CFM can recover from -26% accuracy drops to improve accuracy by +1.6% (row: 6-8). Fig. 7.7 further shows that the improvement from CFM is consistent across all classes on multiple datasets.

**Importance of Adaptive Debiasing.** Pseudo-labeling improves our method by +2% (Tab. 7.3, row: 8,9), and we enhance the pseudo-labeling process with adaptive debiasing (PAD) by our CFM and DCM on top of DebiasPL [65] (row: 9, 10). To further verify the effectiveness of our

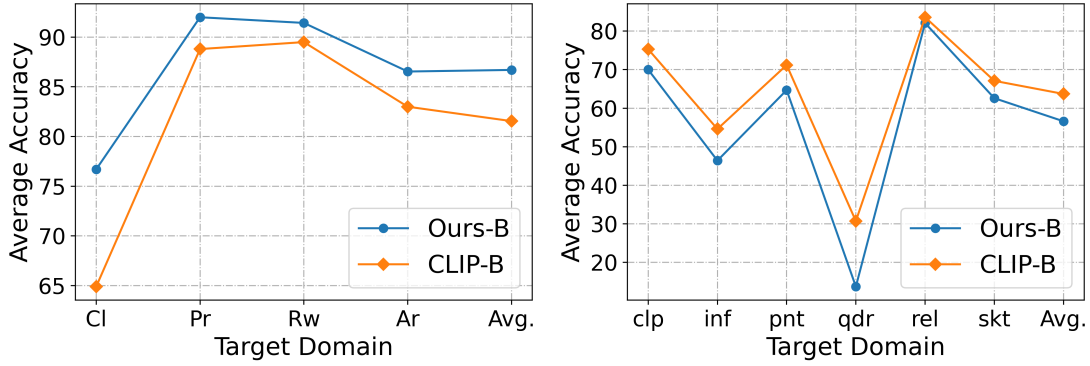


Figure 7.7. Our improvement compared to the original CLIP [59]: the score is averaged over all tasks using each domain as the target domain. “-B” refers to ViT-B [56].

debias method, we summarize the confusion matrix in Fig. 7.5 and compute precision/recall in Fig. 7.3 to show improvement in inter-class confusion (“car”, “bus” and “truck” belong to “vehicle” category).

## 7.5.2 External comparison

We achieved state-of-the-art results on four public datasets using ViT-B backbone, and we test multiple convolutional backbones for a fair comparison.

**VisDA-2017.** We first use ResNet-101 (RN-101) as the baseline model to perform fair comparisons with recent methods [54, 265, 266, 267, 268, 223]. Tab. 7.4 shows that our method consistently improves almost all classes, and improves 4.2% on the average accuracy compared to the previous best method, SDAT [223]. We then follow ViT-based methods [49, 223, 48] to use ViT-B (denote as “-B”) and achieve superior performance compared to the state-of-the-art methods. We also observe a significant accuracy increase on “truck”, thanks to our adaptive debiasing module that makes “truck” more discriminate from “car”.

**Office-Home/31.** For Office-Home [68], we first use ResNet-50 (RN-50) to fairly compare with recent methods [266, 223, 54, 53]. Tab. 7.5 shows a +2.7% increase from the previous best method, KUDA [236]. We then follow ViT-based methods [223, 48, 232, 49] to use ViT-B, and observe consistent improvement across almost all settings. For office-31 [69], we have a similar observation on consistent improvement compared to the recent methods (Tab. 7.7).

**DomainNet.** Previous improvement (SSRT [48] vs SDAT [223]) only achieves +3.1% accuracy on DomainNet [66] because this is the largest dataset, several domains have completely different

Table 7.4: Accuracies (%) on **VisDA-2017**. \*CDTrans uses DeiT-base backbone. “-B” indicates ViT-B backbone.

| Method         | plane | bcycl | bus  | car  | horse | knife | mcycl | person | plant | sktbrd | train | truck | Avg.        |
|----------------|-------|-------|------|------|-------|-------|-------|--------|-------|--------|-------|-------|-------------|
| RN-101 [55]    | 55.1  | 53.3  | 61.9 | 59.1 | 80.6  | 17.9  | 79.7  | 31.2   | 81.0  | 26.5   | 73.5  | 8.5   | 52.4        |
| DANN [47]      | 81.9  | 77.7  | 82.8 | 44.3 | 81.2  | 29.5  | 65.1  | 28.6   | 51.9  | 54.6   | 82.8  | 7.8   | 57.4        |
| CDAN [228]     | 85.2  | 66.9  | 83.0 | 50.8 | 84.2  | 74.9  | 88.1  | 74.5   | 83.4  | 76.0   | 81.9  | 38.0  | 73.9        |
| SAFN [269]     | 93.6  | 61.3  | 84.1 | 70.6 | 94.1  | 79.0  | 91.8  | 79.6   | 89.9  | 55.6   | 89.0  | 24.4  | 76.1        |
| SWD [270]      | 90.8  | 82.5  | 81.7 | 70.5 | 91.7  | 69.5  | 86.3  | 77.5   | 87.4  | 63.6   | 85.6  | 29.2  | 76.4        |
| CaCo [267]     | 90.4  | 80.7  | 78.8 | 57.0 | 88.9  | 87.0  | 81.3  | 79.4   | 88.7  | 88.1   | 86.8  | 63.9  | 80.9        |
| SUDA [268]     | 91.5  | 79.7  | 71.9 | 66.5 | 88.5  | 81.1  | 85.6  | 79.5   | 86.2  | 86.5   | 79.9  | 74.3  | 80.9        |
| DTA [229]      | 93.7  | 82.2  | 85.6 | 83.8 | 93.0  | 81.0  | 90.7  | 82.0   | 95.1  | 78.1   | 86.4  | 32.1  | 81.5        |
| CGDM [265]     | 93.4  | 82.7  | 73.2 | 68.4 | 92.9  | 94.5  | 88.7  | 82.1   | 93.4  | 82.5   | 86.8  | 49.2  | 82.3        |
| SHOT [51]      | 94.3  | 88.5  | 80.1 | 57.3 | 93.1  | 94.9  | 80.7  | 80.3   | 91.5  | 89.1   | 86.3  | 58.2  | 82.9        |
| NWD [54]       | 96.1  | 82.7  | 76.8 | 71.4 | 92.5  | 96.8  | 88.2  | 81.3   | 92.2  | 88.7   | 84.1  | 53.7  | 83.7        |
| SDAT [223]     | 95.8  | 85.5  | 76.9 | 69.0 | 93.5  | 97.4  | 88.5  | 78.2   | 93.1  | 91.6   | 86.3  | 55.3  | 84.3        |
| MSGD [266]     | 97.5  | 83.4  | 84.4 | 69.4 | 95.9  | 94.1  | 90.9  | 75.5   | 95.5  | 94.6   | 88.1  | 44.9  | 84.6        |
| CAN [227]      | 97.0  | 87.2  | 82.5 | 74.3 | 97.8  | 96.2  | 90.8  | 80.7   | 96.6  | 96.3   | 87.5  | 59.9  | 87.2        |
| AaD [230]      | 97.4  | 90.5  | 80.8 | 76.2 | 97.3  | 96.1  | 89.8  | 82.9   | 95.5  | 93.0   | 92.0  | 64.7  | 88.0        |
| Ours-RN        | 96.7  | 88.8  | 87.0 | 82.8 | 97.1  | 93.0  | 91.3  | 83.0   | 95.5  | 91.8   | 91.5  | 63.0  | <b>88.5</b> |
| ViT-B [56]     | 99.1  | 60.7  | 70.6 | 82.7 | 96.5  | 73.1  | 97.1  | 19.7   | 64.5  | 94.7   | 97.2  | 15.4  | 72.6        |
| TVT-B [232]    | 92.9  | 85.6  | 77.5 | 60.5 | 93.6  | 98.2  | 89.4  | 76.4   | 93.6  | 92.0   | 91.7  | 55.7  | 83.9        |
| SHOT-B [49]    | 97.9  | 90.3  | 86.0 | 73.4 | 96.9  | 98.8  | 94.3  | 54.8   | 95.4  | 87.1   | 93.4  | 62.7  | 85.9        |
| CDTrans*[49]   | 97.1  | 90.5  | 82.4 | 77.5 | 96.6  | 96.1  | 93.6  | 88.6   | 97.9  | 86.9   | 90.3  | 62.8  | 88.4        |
| SSRT-B [48]    | 98.9  | 87.6  | 89.1 | 84.8 | 98.3  | 98.7  | 96.3  | 81.1   | 94.9  | 97.9   | 94.5  | 43.1  | 88.8        |
| SDAT-B [223]   | 98.4  | 90.9  | 85.4 | 82.1 | 98.5  | 97.6  | 96.3  | 86.1   | 96.2  | 96.7   | 92.9  | 56.8  | 89.8        |
| PMTTrans [271] | 98.9  | 93.7  | 84.5 | 73.3 | 99.0  | 98.0  | 96.2  | 67.8   | 94.2  | 98.4   | 96.6  | 49.0  | 87.5        |
| Ours-B         | 98.1  | 93.8  | 87.1 | 85.5 | 98.0  | 96.0  | 94.4  | 86.0   | 94.9  | 93.3   | 93.5  | 70.2  | <b>90.9</b> |

appearances (e.g., infographic vs quickdraw), and the distributions among different domains are imbalanced. Our method, however, achieves +18.5% improvement over the previous best method, SSRT, thanks to our proposed CFM for preserving the original CLIP’s pre-trained knowledge, and our pseudo-label with adaptive debiasing for improving the pseudo-label quality.

### 7.5.3 Generalization of our method

**Applications.** We observe catastrophic forgetting issue in CLIP across multiple applications where the accuracy decreases when fine-tuning CLIP in Incremental Learning (-9.9%), and Domain Generalization (-14.0%) and CFM can increase the accuracy +5.3%, +2.4% respectively. For few-shot learning, our method outperforms Tip-Adapter-F [274] by 0.5%.

Table 7.5: Accuracies (%) on **Office-Home**. \*CDTrans uses DeiT-Base backbone. “-B” indicates ViT-B backbone.

| Method         | Ar→Cl | Ar→Pr | Ar→Rw | Cl→Ar | Cl→Pr | Cl→Rw | Pr→Ar | Pr→Cl | Pr→Rw | Rw→Ar | Rw→Cl | Rw→Pr | Avg.        |
|----------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------------|
| RN-50 [55]     | 34.9  | 50.0  | 58.0  | 37.4  | 41.9  | 46.2  | 38.5  | 31.2  | 60.4  | 53.9  | 41.2  | 59.9  | 46.1        |
| CDAN+E [228]   | 50.7  | 70.6  | 76.0  | 57.6  | 70.0  | 70.0  | 57.4  | 50.9  | 77.3  | 70.9  | 56.7  | 81.6  | 65.8        |
| SAFN [269]     | 52.0  | 71.7  | 76.3  | 64.2  | 69.9  | 71.9  | 63.7  | 51.4  | 77.1  | 70.9  | 57.1  | 81.5  | 67.3        |
| CDAN+TN [272]  | 50.2  | 71.4  | 77.4  | 59.3  | 72.7  | 73.1  | 61.0  | 53.1  | 79.5  | 71.9  | 59.0  | 82.9  | 67.6        |
| FGDA+MDD [52]  | 57.1  | 77.5  | 81.0  | 68.4  | 77.2  | 75.9  | 65.8  | 55.8  | 81.0  | 74.3  | 60.5  | 83.6  | 71.5        |
| SHOT [51]      | 57.1  | 78.1  | 81.5  | 68.0  | 78.2  | 78.1  | 67.4  | 54.9  | 82.2  | 73.3  | 58.8  | 84.3  | 71.8        |
| SDAT [223]     | 58.2  | 77.1  | 82.2  | 66.3  | 77.6  | 76.8  | 63.3  | 57.0  | 82.2  | 74.9  | 64.7  | 86.0  | 72.2        |
| MSGD [266]     | 58.7  | 76.9  | 78.9  | 70.1  | 76.2  | 76.6  | 69.0  | 57.2  | 82.3  | 74.9  | 62.7  | 84.5  | 72.4        |
| MCC+NWD [54]   | 58.1  | 79.6  | 83.7  | 67.7  | 77.9  | 78.7  | 66.8  | 56.0  | 81.9  | 73.9  | 60.9  | 86.1  | 72.6        |
| AaD [230]      | 59.3  | 79.3  | 82.1  | 68.9  | 79.8  | 79.5  | 67.2  | 57.4  | 83.1  | 72.1  | 58.5  | 85.4  | 72.7        |
| CST [234]      | 59.0  | 79.6  | 83.4  | 68.4  | 77.1  | 76.7  | 68.9  | 56.4  | 83.0  | 75.3  | 62.2  | 85.1  | 73.0        |
| DCAN+SCDA [53] | 60.7  | 76.4  | 82.8  | 69.8  | 77.5  | 78.4  | 68.9  | 59.0  | 82.7  | 74.9  | 61.8  | 84.5  | 73.1        |
| KUDA [236]     | 58.2  | 80.0  | 82.9  | 71.1  | 80.3  | 80.7  | 71.3  | 56.8  | 83.2  | 75.5  | 60.3  | 86.6  | 73.9        |
| Ours (RN-50)   | 57.5  | 84.0  | 83.8  | 77.8  | 85.5  | 84.7  | 76.3  | 59.2  | 85.4  | 78.1  | 60.2  | 86.7  | <b>76.6</b> |
| ViT-B [56]     | 54.7  | 83.0  | 87.2  | 77.3  | 83.4  | 85.5  | 74.4  | 50.9  | 87.2  | 79.6  | 53.8  | 88.8  | 75.5        |
| SHOT-B [49]    | 67.1  | 83.5  | 85.5  | 76.6  | 83.4  | 83.7  | 76.3  | 65.3  | 85.3  | 80.4  | 66.7  | 83.4  | 78.1        |
| CDTrans*[49]   | 68.8  | 85.0  | 86.9  | 81.5  | 87.1  | 87.3  | 79.6  | 63.3  | 88.2  | 82.0  | 66.0  | 90.6  | 80.5        |
| TVT-B [232]    | 74.9  | 86.8  | 89.5  | 82.8  | 88.0  | 88.3  | 79.8  | 71.9  | 90.1  | 85.5  | 74.6  | 90.6  | 83.6        |
| SDAT-B [223]   | 70.8  | 87.0  | 90.5  | 85.2  | 87.3  | 89.7  | 94.1  | 70.7  | 90.6  | 88.3  | 75.5  | 92.1  | 84.3        |
| SSRT-B [48]    | 75.2  | 89.0  | 91.1  | 85.1  | 88.3  | 89.9  | 85.0  | 74.2  | 91.3  | 85.7  | 78.6  | 91.8  | 85.4        |
| PMTans [271]   | 81.2  | 91.6  | 92.4  | 88.9  | 91.6  | 93.0  | 88.5  | 80.0  | 93.4  | 89.5  | 82.4  | 94.5  | <u>88.9</u> |
| Ours-B         | 76.4  | 90.6  | 90.8  | 86.7  | 92.3  | 92.0  | 86.0  | 74.5  | 91.5  | 86.9  | 79.1  | 93.1  | <b>86.7</b> |

**Vision backbone.** Our method works on both vision language (CLIP), and vision model. With our method, BiT-M-R101x3 [275] (trained on JFT-300M) achieved 88.1% on VisDA-2017 (comparable to CLIP (90.9%, Tab. 7.4)). However, a small pre-trained dataset is a limitation, as our method has no effect on ResNet-101 with ImageNet-1K due to the minimal effect of catastrophic forgetting issues.

## 7.5.4 Computational complexity

It takes 16.5 hours to train CLIP with ViT-B backbone on an Nvidia Tesla V100 GPU for VisDA-2017 [67]. Pseudo-label is a standard method in UDA [62, 63, 49, 48] and adding pseudo-label increases the training times to 23.3 hours. To compute CFM, we forward the entire dataset with the original CLIP (0.6 hours for VisDA-2017) to cache the original CLIP representation (only needs to do once). We did not observe any overhead from CFM during training since CFM simply compare the low dimensional representations (obtain as part of the training). We observe

Table 7.6: Accuracies (%) on **DomainNet**. In each sub-table, the column-wise means source domain and the row-wise means target domain. “-B” indicates ViT-B (except CDTrans uses DeiT).

| <b>ResNet-101 [55]</b> | clp  | inf  | pnt  | qdr  | rel  | skt  | Avg. | <b>MIMTFL [273]</b> | clp  | inf  | pnt  | qdr  | rel  | skt  | Avg. | <b>CDAN [228]</b> | clp  | inf  | pnt  | qdr | rel  | skt  | Avg. |
|------------------------|------|------|------|------|------|------|------|---------------------|------|------|------|------|------|------|------|-------------------|------|------|------|-----|------|------|------|
| clp                    | -    | 19.3 | 37.5 | 11.1 | 52.2 | 41.0 | 32.2 | clp                 | -    | 15.1 | 35.6 | 10.7 | 51.5 | 43.1 | 31.2 | clp               | -    | 20.4 | 36.6 | 9.0 | 50.7 | 42.3 | 31.8 |
| inf                    | 30.2 | -    | 31.2 | 3.6  | 44.0 | 27.9 | 27.4 | inf                 | 32.1 | -    | 31.0 | 2.9  | 48.5 | 31.0 | 29.1 | inf               | 27.5 | -    | 25.7 | 1.8 | 34.7 | 20.1 | 22.0 |
| pnt                    | 39.6 | 18.7 | -    | 4.9  | 54.5 | 36.3 | 30.8 | pnt                 | 40.1 | 14.7 | -    | 4.2  | 55.4 | 36.8 | 30.2 | pnt               | 42.6 | 20.0 | -    | 2.5 | 55.6 | 38.5 | 31.8 |
| qdr                    | 7.0  | 0.9  | 1.4  | -    | 4.1  | 8.3  | 4.3  | qdr                 | 18.8 | 3.1  | 5.0  | -    | 16.0 | 13.8 | 11.3 | qdr               | 21.0 | 4.5  | 8.1  | -   | 14.3 | 15.7 | 12.7 |
| rel                    | 48.4 | 22.2 | 49.4 | 6.4  | -    | 38.8 | 33.0 | rel                 | 48.5 | 19.0 | 47.6 | 5.8  | -    | 39.4 | 32.1 | rel               | 51.9 | 23.3 | 50.4 | 5.4 | -    | 41.4 | 34.5 |
| skt                    | 46.9 | 15.4 | 37.0 | 10.9 | 47.0 | -    | 31.4 | skt                 | 51.7 | 16.5 | 40.3 | 12.3 | 53.5 | -    | 34.9 | skt               | 50.8 | 20.3 | 43.0 | 2.9 | 50.8 | -    | 33.6 |
| Avg.                   | 34.4 | 15.3 | 31.3 | 7.4  | 40.4 | 30.5 | 26.6 | Avg.                | 38.2 | 13.7 | 31.9 | 7.2  | 45.0 | 32.8 | 28.1 | Avg.              | 38.8 | 17.7 | 32.8 | 4.3 | 41.2 | 31.6 | 27.7 |

| <b>MDD+SCDA [53]</b> | clp  | inf  | pnt  | qdr  | rel  | skt  | Avg. | <b>ViT-B [56]</b> | clp  | inf  | pnt  | qdr  | rel  | skt  | Avg. | <b>CD-Trans [49]</b> | clp  | inf  | pnt  | qdr  | rel  | skt  | Avg. |
|----------------------|------|------|------|------|------|------|------|-------------------|------|------|------|------|------|------|------|----------------------|------|------|------|------|------|------|------|
| clp                  | -    | 20.4 | 43.3 | 15.2 | 59.3 | 46.5 | 36.9 | clp               | -    | 27.2 | 53.1 | 13.2 | 71.2 | 53.3 | 43.6 | clp                  | -    | 29.4 | 57.2 | 26.0 | 72.6 | 58.1 | 48.7 |
| inf                  | 32.7 | -    | 34.5 | 6.3  | 47.6 | 29.2 | 30.1 | inf               | 51.4 | -    | 49.3 | 4.0  | 66.3 | 41.1 | 42.4 | inf                  | 57.0 | -    | 54.4 | 12.8 | 69.5 | 48.4 | 48.4 |
| pnt                  | 46.4 | 19.9 | -    | 8.1  | 58.8 | 42.9 | 35.2 | pnt               | 53.1 | 25.6 | -    | 4.8  | 70.0 | 41.8 | 39.1 | pnt                  | 62.9 | 27.4 | -    | 15.8 | 72.1 | 53.9 | 46.4 |
| qdr                  | 31.1 | 6.6  | 18.0 | -    | 28.8 | 22.0 | 21.3 | qdr               | 30.5 | 4.5  | 16.0 | -    | 27.0 | 19.3 | 19.5 | qdr                  | 44.6 | 8.9  | 29.0 | -    | 42.6 | 28.5 | 30.7 |
| rel                  | 55.5 | 23.7 | 52.9 | 9.5  | -    | 45.2 | 37.4 | rel               | 58.4 | 29.0 | 60.0 | 6.0  | -    | 45.8 | 39.9 | rel                  | 66.2 | 31.0 | 61.5 | 16.2 | -    | 52.9 | 45.6 |
| skt                  | 55.8 | 20.1 | 46.5 | 15.0 | 56.7 | -    | 38.8 | skt               | 63.9 | 23.8 | 52.3 | 14.4 | 67.4 | -    | 44.4 | skt                  | 69.0 | 29.6 | 59.0 | 27.2 | 72.5 | -    | 51.5 |
| Avg.                 | 44.3 | 18.1 | 39.0 | 10.8 | 50.2 | 37.2 | 33.3 | Avg.              | 51.5 | 22.0 | 46.1 | 8.5  | 60.4 | 40.3 | 38.1 | Avg.                 | 59.9 | 25.3 | 52.2 | 19.6 | 65.9 | 48.4 | 45.2 |

| <b>PMTrans [271]</b> | clp  | inf  | pnt  | qdr  | rel  | skt  | Avg. | <b>SSRT-B [48]</b> | clp  | inf  | pnt  | qdr  | rel  | skt  | Avg. | <b>Ours-B</b> | clp  | inf  | pnt  | qdr  | rel  | skt  | Avg. |
|----------------------|------|------|------|------|------|------|------|--------------------|------|------|------|------|------|------|------|---------------|------|------|------|------|------|------|------|
| clp                  | -    | 34.2 | 62.7 | 32.5 | 79.3 | 63.7 | 54.5 | clp                | -    | 33.8 | 60.2 | 19.4 | 75.8 | 59.8 | 49.8 | clp           | -    | 73.6 | 75.4 | 74.6 | 76.4 | 76.3 | 75.3 |
| inf                  | 67.4 | -    | 61.1 | 22.2 | 78.0 | 57.6 | 57.3 | inf                | 55.5 | -    | 54.0 | 9.0  | 68.2 | 44.7 | 46.3 | inf           | 55.1 | -    | 54.3 | 53.6 | 54.9 | 54.9 | 54.6 |
| pnt                  | 69.7 | 33.5 | -    | 23.9 | 79.8 | 61.2 | 53.6 | pnt                | 61.7 | 28.5 | -    | 8.4  | 71.4 | 55.2 | 45.0 | pnt           | 71.1 | 70.6 | -    | 70.0 | 72.7 | 71.7 | 71.2 |
| qdr                  | 54.6 | 17.4 | 38.9 | -    | 49.5 | 41.0 | 40.3 | qdr                | 42.5 | 8.8  | 24.2 | -    | 37.6 | 33.6 | 29.3 | qdr           | 36.8 | 18.0 | 32.0 | -    | 31.7 | 34.9 | 30.7 |
| rel                  | 74.1 | 35.3 | 70.0 | 25.4 | -    | 61.1 | 53.2 | rel                | 69.9 | 37.1 | 66.0 | 10.1 | -    | 58.9 | 48.4 | rel           | 84.2 | 83.5 | 83.5 | 83.1 | -    | 83.6 | 83.6 |
| skt                  | 73.8 | 33.0 | 62.6 | 30.9 | 77.5 | -    | 55.6 | skt                | 70.6 | 32.8 | 62.2 | 21.7 | 73.2 | -    | 52.1 | skt           | 68.1 | 66.6 | 67.2 | 66.1 | 67.5 | -    | 67.1 |
| Avg.                 | 67.9 | 30.7 | 59.1 | 27.0 | 72.8 | 56.9 | 52.4 | Avg.               | 60.0 | 28.2 | 53.3 | 13.7 | 65.3 | 50.4 | 45.2 | Avg.          | 63.1 | 62.5 | 62.5 | 69.5 | 60.6 | 64.3 | 63.7 |

a training time increase to 23.5 hours from our adaptive debiasing, but the overhead is trivial with  $<1\%$  increases from the pseudo-label setting. Our method does not change the test speed.

## 7.6 Conclusion

We propose CLIP in the UDA setting. We first include a domain name into a prompt, and we uncover catastrophic forgetting issues when fine-tuning CLIP. We propose to counter this by adjusting the learning rate according to CFM. We add pseudo-labeling by further extending DebiasPL (from a single domain to multiple domains in the UDA setting) with our CFM and DCM to better adjust debias strength and momentum. Our method does not introduce

Table 7.7: Accuracies (%) on **Office-31**.

| Method         | A→W  | D→W  | W→D   | A→D  | D→A  | W→A  | Avg.        |
|----------------|------|------|-------|------|------|------|-------------|
| RN-50 [55]     | 68.4 | 96.7 | 99.3  | 68.9 | 62.5 | 60.7 | 76.1        |
| DANN [47]      | 82.0 | 96.9 | 99.1  | 79.7 | 68.2 | 67.4 | 82.2        |
| SAFN+ENT [269] | 90.1 | 98.6 | 99.8  | 90.7 | 73.0 | 70.2 | 87.1        |
| SUDA [268]     | 90.8 | 98.7 | 100.0 | 91.2 | 72.2 | 71.4 | 87.4        |
| CaCo [267]     | 89.7 | 98.4 | 100.0 | 91.7 | 73.1 | 72.8 | 87.6        |
| SHOT [51]      | 90.1 | 98.4 | 99.9  | 94.0 | 74.7 | 74.3 | 88.6        |
| CDAN+TN [272]  | 95.7 | 98.7 | 100.  | 94.0 | 73.4 | 74.2 | 89.3        |
| MDD+SCDA [53]  | 95.3 | 99.0 | 100.  | 95.4 | 77.2 | 75.9 | 90.5        |
| ViT-B [56]     | 91.2 | 99.2 | 100.  | 90.4 | 81.1 | 80.6 | 90.4        |
| SHOT-B [51]    | 94.3 | 99.0 | 100.  | 95.3 | 79.4 | 80.2 | 91.4        |
| CDTrans* [49]  | 96.7 | 99.0 | 100.  | 97.0 | 81.1 | 81.9 | 92.6        |
| SSRT-B [48]    | 97.7 | 99.2 | 100.  | 98.6 | 83.5 | 82.2 | 93.5        |
| TVT-B [232]    | 96.4 | 99.4 | 100.  | 96.4 | 84.9 | 86.1 | 93.8        |
| PMTrans [271]  | 99.1 | 99.6 | 100.0 | 99.4 | 85.7 | 86.3 | <u>95.0</u> |
| Ours-B         | 97.9 | 99.2 | 100.  | 98.5 | 84.6 | 85.3 | 94.3        |

computational overhead, and achieves superior results than the state-of-the-art methods on four public datasets, with a large improvement (+18.5%) on DomainNet. For future work, CFM and DCM could be improved to a more sophisticated function or even learnable.



# Chapter 8

## Path-CLIP: Efficient Adaptation of CLIP for Pathology Image Analysis with Limited Data

Contrastive Language-Image Pre-training (CLIP) has shown its ability of learning distinctive visual representations and generalization to various downstream vision tasks. However, its applicability in pathology image analysis with limited labeled data is still under study due to the giant domain shift (natural photos vs pathology images) and catastrophic forgetting issues. Furthermore, pathological tasks may need to be separated into multiple learning tasks. Hence, it is critical to study efficient adaptation schemes in this domain for scalable analysis. In this work, we propose Path-CLIP to quickly adapt CLIP to multiple pathology tasks. First, we propose Residual Feature Connection (RFC) with a self-adaptive ratio to fuse and balance the source and task-specific knowledge. Second, we propose Hidden Representation Perturbation (HRP) and Dual-view Vision Contrastive (DVC) to alleviate the overfitting issue. Lastly, we propose Doublet Multimodal Contrastive Loss (DMCL) to adapt CLIP to pathology tasks. We show Path-CLIP can adapt pre-trained CLIP to downstream pathology tasks and achieve competitive results. Specifically, Path-CLIP achieves over 19% improvement in accuracy when only using 0.1% of labeled data in PCam with only 10 minutes of fine-tuning while running on a single GPU.

## 8.1 Introduction

Deep learning with better network designs and large-scale well-curated datasets has achieved significant performance improvement in pathology image analysis tasks [36, 29]. However, collecting high-quality datasets with reliable annotations for each individual vision task can be time-consuming and labor-intensive [71, 72]. This may prevent the broad adoption of advanced deep learning techniques. To relieve the reliance on such datasets, pre-training and fine-tuning methods (TFT) have been studied in vision tasks: pre-train the model on a large-scale dataset and then fine-tune the model on different downstream tasks [73]. There are several challenges of such methods: 1) they may still require a large amount of labeled set to avoid the overfitting issue when fine-tuning the model for the downstream task [74, 75]; 2) the fine-tuning may not bring satisfactory performance in the target domain due to the existence of a large domain gap between the pre-trained data and pathology images [76]. Therefore, a natural question is how to efficiently fit the model to the new task with minimal overfitting when the new dataset is small.

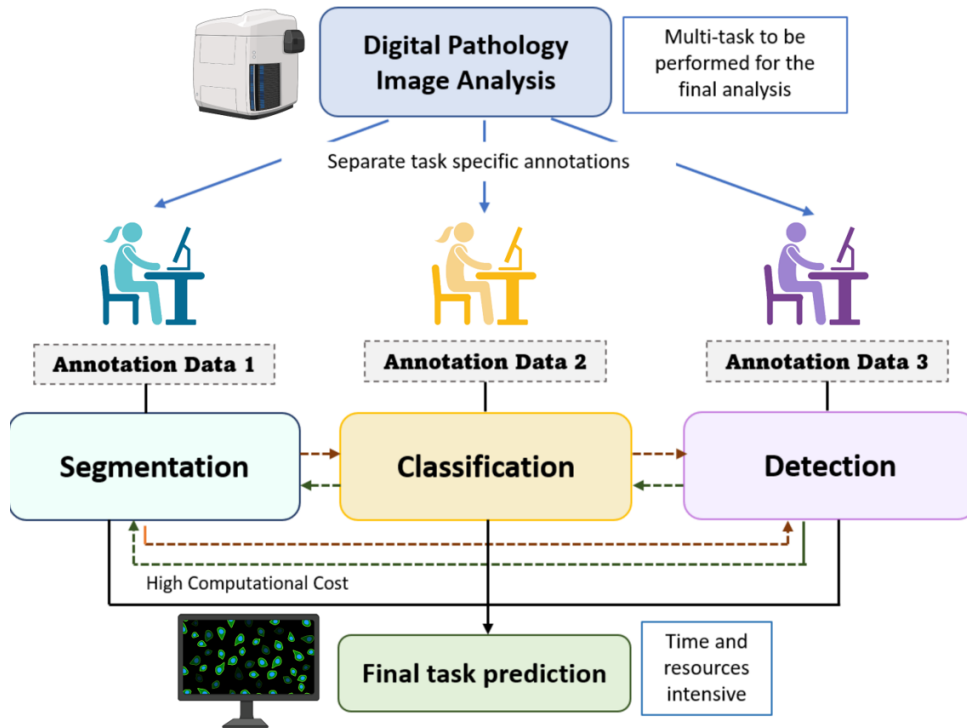


Figure 8.1. Overview of the traditional computer vision training in medical image analysis task.

To fill the performance gap due to domain shift, Contrastive Language-Image Pre-training

(CLIP) [59] has shown its power in learning generic and distinctive visual representations via language supervision. It aligns images and texts in the same feature space and uses a contrastive loss to formulate the learning objective. CLIP uses two separate encoders for images and texts, then maximizes the similarity score of positive pairs of images and texts while minimizing it of negative pairs [59, 241]. It achieves promising results on various image classification tasks without needing any annotated data, i.e., zero-shot transfer settings. As a language-vision model, CLIP uses prompts as the supervision, where the visual labels are entered into the hand-crafted template. By pre-training the model at a large scale, models can learn the visual contents and easily be transferred to downstream tasks through the prompt-based zero-shot transfer.

However, the manual design of prompts can be a non-trivial and time-consuming task. In [241], the authors found that even a slight change in the prompt (e.g., one word) can make a big difference. They introduced Context Optimization (CoOp) to automate prompt engineering to generate continuous soft prompts instead of using hand-chosen hard prompts [241]. However, CoOp requires substantial computing resources and the results of CoOp are not interpretable. Furthermore, CoOp faces performance degradation when there is a significant domain shift, e.g., from natural images to pathology images, making it hard to be adapted to medical imaging tasks. On the other hand, medical image analysis may involve multiple learning tasks simultaneously. It is impractical if the fine-tuning requires heavy computational resources when the learning tasks are separated from the analysis objective. As shown in Figure 8.1, multi-tasks need to be performed for the final analysis of the digital pathology domain: this may involve segmentation, classification, and/or detection tasks. For example, a recent work [29] quantifies and visualizes each type of pathology in grey and white matter regions in gigapixel images, respectively. To achieve such an objective, we need to separate it into two learning tasks: pathology detection and grey/white matter (GM/WM) segmentation. They collected and annotated two datasets, then trained two models from scratch to achieve it. Therefore, it will be extremely beneficial if we can propose an effective way to adapt a big model into multiple downstream tasks quickly so the final analysis objective can be reached in a reasonable time manner.

In this work, we aim to fine-tune CLIP efficiently with light computing resources for pathology image analysis tasks. There are several challenges in fine-tuning CLIP. First, overfitting

is a severe issue if we adapt CLIP to the downstream tasks directly since CLIP is pre-trained on a 400M dataset while the new domain dataset can be small [77]. Second, CLIP has shown its promising zero-shot generalization ability, directly fine-tuning it may lose it and face the “catastrophic forgetting” issue. In other words, it is unclear how to effectively learn the new knowledge while retaining the original pre-trained knowledge so that the generalization of CLIP can be maintained. Third, there are limited studies on benchmarking CLIP’s transferability in the pathology domain; hence, its applicability remains unclear. To address the above issues, we first study the applicability of CLIP on two public pathology datasets and benchmark the zero-shot ability of CLIP on them. Then we did empirical experiments to verify the importance of the CLIP’s pre-trained knowledge. To address the “catastrophic forgetting” issue, we propose Residual Feature Connection (RFC) as a lightweight approach for adapting CLIP to pathology images. In RFC, we design a self-adaptive module to automatically fuse/balance the original knowledge from CLIP and the new knowledge learned from the new pathological task with only a few additional trainable weights instead of optimizing the entire encoders in CLIP. To further alleviate the overfitting, we also incorporate Hidden Representation Perturbation (HRP) and Dual-view Vision Contrastive (DVC). Lastly, we build an end-to-end adaptation of CLIP with a proposed loss, Doublet Multimodal Contrastive Loss (DMCL), to pathology tasks, named Path-CLIP.

We summarize our contributions as follows:

- We explore the applicability of CLIP on pathology images and highlight the catastrophic forgetting issues when adapting CLIP to downstream pathology tasks
- We propose Path-CLIP to adapt CLIP to pathology tasks with limited data. In Path-CLIP, we propose Residual Feature Connection (RFC) with Hidden Representation Perturbation (HRP) and a self-adaptive ratio to alleviate the overfitting and catastrophic forgetting issues. We also propose Dual-view Vision Contrastive (DVC) and Doublet Multimodal Contrastive Loss (DMCL) to fine-tune CLIP as an end-to-end pipeline.
- Extensive experiments demonstrate the efficiency and robustness of Path-CLIP. It has the potential to quickly adapt pre-trained CLIP to multi-tasks with competitive performance

and light computational cost.

## 8.2 Related work

### 8.2.1 Language-vision model

Language-vision models have shown promising performance in learning generic visual representations [59, 237, 238, 239]. Recent models advance via text representation learning with large-scale Transformers [240] and even web-scale training datasets [241]. Transformer-based multimodal learning has achieved great success in such large-scale datasets [276, 277]. For example, CLIP [59] was trained on 400 million image-caption pairs and achieved state-of-the-art results in many fields [58, 59, 60, 61]. In CLIP’s architecture, it has two encoders: the vision encoder can be ResNet [55] or ViT [278] while the text encoder can be Transformer (e.g., BERT [279]). A recent work [280] fine-tuned CLIP for video data and performed competitively to more complicated methods designed for video processing. PointCLIP [281] applied CLIP to 3D recognition. CLIP is also applied for image generation tasks [282] and shows its power to reduce data collection. However, the best way to adapt CLIP for downstream tasks is still being studied, especially when the new domain is far from the pre-trained domains, e.g., the medical domain.

### 8.2.2 Language-vision training in medical domain

Language-vision pre-training typically requires near-infinite web images and captions from general domains, e.g., CLIP uses 400 million image-caption pairs [59]. Such datasets dwarf the scale of medical datasets. Another challenge is the annotation cost: medical images typically require domain knowledge from trained personnel [72]. This makes the training cost-expensive if we have multiple medical tasks. For example, Lai *et al.* studies the distribution of Amyloid- $\beta$  plaques (a hallmark pathology with Alzheimer disease) in grey and white matter, which involves two learning tasks: image segmentation and object detection. These two tasks require two different datasets with their required annotation to train two models [29] (more details can be found in Chapter 2).

Although expert annotation of medical images can be expensive, preparing the captions and prompts for images will exacerbate the situation if we use language-vision training. For example,

MedCLIP [283] used 570 thousands of image-text pairs to achieve 60% zero-shot accuracy. Datasets with such scale are challenging to curate in pathology image tasks, where each slide is at gigapixel level. The efforts to adapt CLIP for the pathology domain remain limited due to the large gap between the general and medical image domains. In this work, we aim to design an efficient adaption framework that can be scalable for multiple downstream tasks.

### 8.2.3 CLIP fine-tuning and adaptation

Deep learning methods are data-hungry, which can prevent their wide adoption in real-world scenarios. Pre-training and fine-tuning offer a promising direction for adapting the model to downstream tasks. Although recent multimodal models (e.g., CLIP [59]) perform astonishing standard vision tasks without fine-tuning, the gap between CLIP and supervised learning is still noticeable. Therefore, many works focus on fine-tuning CLIP with a few samples from the new task provided. For example, CoOp [241] kept the entire pre-trained parameters fixed but optimized the continuous prompt through a learnable vector. This method closes the gap between CLIP and supervised learning but the performance gain is limited as it only focuses on the perspective of prompt design instead of the vision part for the image recognition tasks. On the other hand, CLIP-Adapter [73] and Tip-Adapter [274] freeze the pre-trained weights and train additive module, which significantly improves the CLIP’s performance with limited training data in the new task. These methods require a lower computational cost. However, both CLIP-Adapter [73] and Tip-Adapter [274] involve tuning a ratio to balance the original and fine-tuned knowledge manually, which may not be practical when we have multiple target learning tasks in the medical domain. Therefore, in this work, we propose a scalable approach to automate this process.

## 8.3 Methodology

In this section, we introduce the proposed Path-CLIP. We summarize the end-to-end pipeline in Fig. 8.2: a self-adaptive residual ratio  $\alpha$  will be incorporated into Residual Feature Connection to fuse and balance the original knowledge with the task-specific knowledge from the new downstream task. We introduce Hidden Representation Perturbation (HRP) and Dual-view Vision Contrastive to alleviate the overfitting issue. Finally, we propose Doublet Multimodal

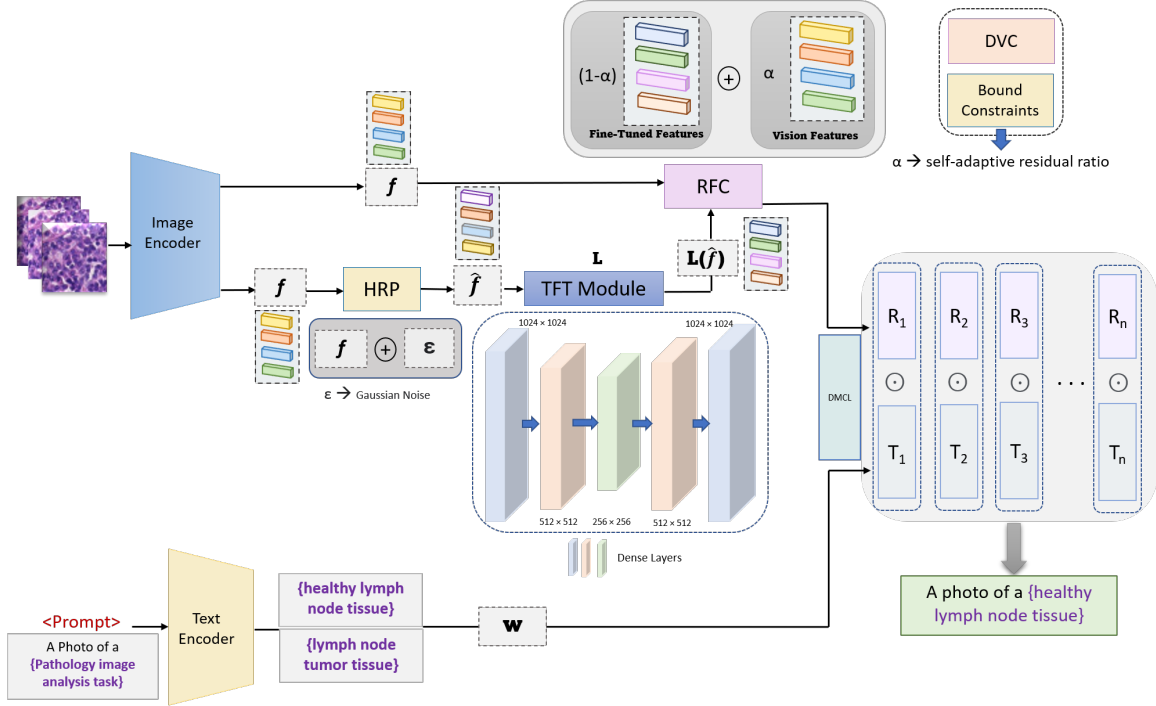


Figure 8.2. Overview of Path-CLIP: the image and text encoders are frozen while RFC (Residual Feature Connection) is a downsampling-upsampling architecture of fully connected layers. RFC blends the fine-tuned knowledge with the original knowledge from CLIP’s vision encoder ( $F(\cdot)$ ) via a self-adaptive residual ratio  $\alpha$ . HRP refers to Hidden Representation Perturbation.

Contrastive Loss (DMCL) as the loss function for fine-tuning.

### 8.3.1 Language-Vision Pre-training

We first revisit language-vision pre-training and use CLIP [59] as one example. Note that our framework is applicable to broader language-vision models. CLIP [59] has a vision encoder  $F(\cdot)$  and a text encoder  $G(\cdot)$ . The vision encoder maps a high-dimensional image into low-dimensional image embeddings. The vision encoder can be a CNN like ResNet [55] or a Vision Transformer (ViT) [278]. The text encoder is built on Transformer [240] and generates text embeddings from the prompt. Specifically, “prompt” here refers to a sequence of words (tokens), such as “a neuropathological image of grey matter”. We collect annotations from experts and then convert them into these prompts.

**Training.** During training, CLIP jointly trains  $F(\cdot)$  and  $G(\cdot)$  to optimize the similarity score (e.g., symmetric cross-entropy loss [256]) between the visual and textual embeddings for each batch. The input consists of an image and its corresponding prompt (e.g., “this is a photo of

grey matter”). They formulate the learning objective as a contrastive loss to align the visual and textual embedding spaces. Given a batch of image-prompt pairs, CLIP maximizes the similarity score for positive pairs while minimizing it for negative pairs in that batch. To learn diverse and wide visual concepts, CLIP’s team collects a dataset of 400 million image-prompt pairs to achieve better transferability to downstream tasks.

**Contrastive loss function.** In CLIP training, the agreement between a positive pair of vision and text embeddings is maximized by InfoNCE loss [284]:

$$l_i^{v \rightarrow t} = -\log \frac{\exp(\mathbf{sim}(v_i, t_i)/\gamma)}{\sum_{k=1}^B \exp(\mathbf{sim}(v_i, t_k)/\gamma)}, \quad (8.1)$$

where  $v, t$  are normalized vectors from vision and text encoders.  $(v_i, t_i)$  is a positive pair.  $\mathbf{sim}$  is a function to measure the similarity of two vectors.  $B$  is the mini-batch size of the image-text pairs and  $\gamma$  is the learnable temperature. Therefore, there is only 1 positive text for the input  $i$ th image but  $N - 1$  negative texts.  $l_i^{v \rightarrow t}$  means the InfoNCE loss from an image  $i$  to texts, so as the text loss  $l_i^{t \rightarrow v}$ . Then the loss function for CLIP is as follows:

$$\mathcal{L}_{\text{CLIP}} = \frac{1}{2B} \sum_{i=1}^B (l_i^{v \rightarrow t} + l_i^{t \rightarrow v}). \quad (8.2)$$

**Inference for classification tasks.** For the inference process, an image  $x$  is transformed into a feature manifold  $f \in \mathbb{R}^D$ , where  $D$  is the feature dimension. Then,  $f$  is multiplied with a classifier weight matrix  $\mathbf{W} \in \mathbb{R}^{D \times K}$ , where  $K$  is the number of classes in the learning task. We get a  $K$ -dimensional logit after matrix multiplication. Then we apply softmax to convert this logit into a probability vector  $p \in \mathbb{R}^K$  over the  $K$  classes. The whole process can be summarized as the following equation:

$$p_i = \frac{\exp(\mathbf{W}_i^T \times f)/\gamma}{\sum_{i=1}^K \exp(\mathbf{W}_i^T \times f)/\gamma}, \quad (8.3)$$

where  $\gamma$  is the temperature parameter learned by CLIP during training and  $\mathbf{W}_i$  is the prototype weight vector for class- $i$ .

**Transferability.** Compared to traditional classifier learning methods where the supervision is a fixed one-hot vector, language-vision pre-training utilizes an open-set visual concept that can be explored through the text encoder to learn a broader semantic space, which in turn makes the learned representations more diverse and adaptable to various downstream tasks (e.g., zero-shot



learning). Therefore, we argue that such language-vision pre-training is helpful for medical problems where the domain is very distinctive from the natural images.

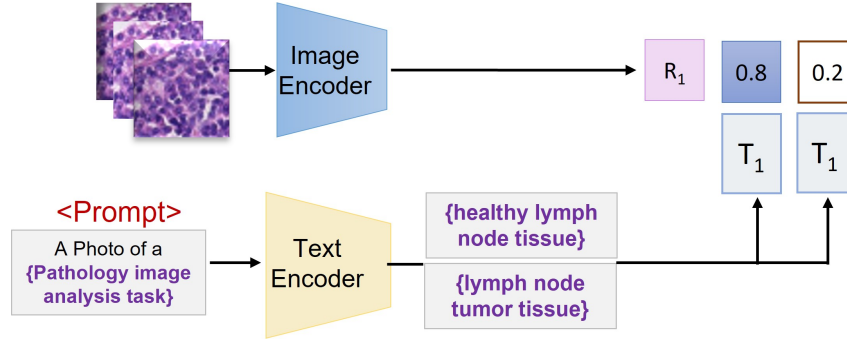


Figure 8.3. Zero-shot pathology image classification.

### 8.3.2 Residual Feature Connection

In this subsection, we introduce Residual Feature Connection (RFC) to learn the task-relevant context when we adapt CLIP to the downstream learning tasks. Unlike CoOp’s prompt tuning, which may not address the domain shift issue between the natural and pathology images, we focus on fine-tuning the visual features  $f$ . However, simple fine-tuning of the entire network may fail in a new pathological task due to 1) the overfitting issue caused by a large amount of parameters in CLIP and a shortage of training samples [285]; 2) catastrophic forgetting issue.

**Importance of the CLIP’s pre-trained knowledge.** CLIP is pre-trained on 400 million of image-text pairs and has shown its zero-shot ability and transferability to a wide range of downstream tasks. We argue preserving pre-trained knowledge of CLIP is crucial to maintain its generalization and alleviate the overfitting issue when the new dataset is small. There are two reasons behind this. First, one recent NLP work [286] shows large pre-trained language models (PLMs) are well-calibrated on the masked language modeling tasks. However, such property cannot be retained through direct fine-tuning due to catastrophic forgetting. Specifically, the pre-trained features are only used for the initialization and will be distorted by the fine-tuning [286]. Inspired by this, we investigate whether this issue also appears in language-vision models. As shown in Table 8.1, "no fine-tuning" but zero-shot gets the best performance in these datasets compared to fin-tune CLIP on 1% of training data in PCam [221]. We also attempt to lower the learning rate to  $1 \times 10^{-6}$ , but the results are still far from satisfactory. Therefore,

we conclude direct fine-tuning will distort the pre-trained knowledge and result in a severe catastrophic forgetting issue.

Besides the catastrophic forgetting issue, overfitting is another common challenge when adapting a big model to a new task. Recent work shows that CLIP’s original knowledge is the key to preserving its generalization [59]. Therefore, to alleviate the above two issues, we propose a residual connection architecture to preserve the pre-trained features and acquire rich semantic information for pathology image analysis.

Table 8.1: Baseline. We fine-tune CLIP (ResNet-50) on PCam [221] with 1% training data. “No fine-tuning” is the best baseline due to the catastrophic forgetting issue.

| # | Configuration                    | Accuracy |
|---|----------------------------------|----------|
| 1 | No fine-tuning                   | 56.5%    |
| 2 | Fine-tune 10 epochs              | 40.3%    |
| 3 | Fine-tune 100 epochs             | 35.9%    |
| 4 | With a lower learning rate on #3 | 45.6%    |

**Architecture overview.** Inspired by [286] that preserving the pre-trained knowledge help calibrate the language models in NLP tasks, we argue the importance of retaining the features from CLIP for computer vision tasks and propose a residual connection architecture to dynamically fuse the fine-tuned knowledge with the original CLIP’s feature. As shown in Fig. 8.2, given an image  $x$ , we get the visual feature  $f$  from the image encoder and compute the classifier weight  $\mathbf{W}$  from the text encoder. Then we design trainable adaptive layers  $L$  to convert  $f$  into  $L(f)$ .  $L$  can be multiple layers of linear transformations in a "down-and-up" architecture. As shown in Fig. 8.2, we have four layers to transform the feature dimension as “1024-256-64-256-1024”. This is one example: the layers are flexible with other dimensions.

After this architecture,  $L(f)$  can be of the same size of  $f$  and blended with  $f$  as follows:

$$f_i^* = (1 - \alpha_i)L(f_i) + \alpha_i f_i, \quad (8.4)$$

where,  $i = \{1, 2, 3, ..n\}$  and  $\alpha$  is the self-adaptive residual ratio to balance the fine-tuned knowledge and the original CLIP’s knowledge. Specifically, the first term adaptively learns

knowledge from the new dataset, while the second term preserves the original knowledge from CLIP.

Then we adopt Eq. (8.3) with the new  $f^*$  to get the class probability vector and predict the category with the highest probability. During the fine-tuning, the weights of the trainable layers are optimized through the symmetric contrastive loss used in CLIP [59]. The prompt can be “this is a photo of [ ]”, where “[ ]” is filled with the class name. For example, “[ ]” can be “healthy lymph node tissue” or “healthy lymph tumor tissue” in PCam [221]. We can incorporate pathological descriptions into the prompt to provide more domain information, e.g., “this is a **pathological breast** photo of [ ]”.

**Hidden Representation Perturbation (HRP).** When we fine-tune a pre-trained language-vision model on a small dataset, overfitting or representation collapse can be a serious issue [287]. To alleviate this issue, we propose Hidden Representation Perturbation (HRP) in this subsection to complement the trainable layers. We select a “four-layer” architecture with HRP instead of more profound architectures so that the number of trainable parameters can remain small. More trainable parameters may tend to overfit the downstream dataset. This issue can be aggravated in limited-resource scenarios, e.g., only a few data samples are available in the new task [288].

To alleviate the overfitting issue in low-resource settings, we focus on adding noise during the training process, a widely-used NLP strategy for large language models to smoothen the optimization landscape. For example, NoisyTune [289] adds the perturbations to the pre-trained weights to improve the model’s generalization. Inspired by these NLP strategies [289, 287], we propose to add random noise to the feature  $f$  before trainable adapting layers fine-tuning it. The random noise can be Gaussian noise  $\varepsilon$  generated from  $\mathcal{N}(0, \sigma^2)$ . After this, the perturbed feature  $f$  is converted into  $\hat{f} = f + \varepsilon$ . Therefore, Eq. (8.4) is converted into:

$$f_i^* = (1 - \alpha_i)L(\hat{f}_i) + \alpha_i f_i. \quad (8.5)$$

Therefore, the residual feature that must be fine-tuned will be perturbed with the Gaussian noise, designed to alleviate the overfitting issue.

### 8.3.3 Self-adaptive residual ratio

In the above section, we introduce residual feature connection and the residual ratio  $\alpha$ .  $\alpha$  helps modify the degree of preserving the original knowledge for better performance. Empirically,  $\alpha$  should be small if the domain gap between the source data and the downstream task is enormous, so more knowledge from the downstream task ( $L(f)$ ) can be considered. However, if  $\alpha$  is too small, the model can be overfitting as it will heavily rely on the fine-tuned features and disregard the pre-trained knowledge from CLIP. On the other hand, the model can also be underfitting if  $\alpha$  is too large, as it may prevent learning from the fine-tuned features. Therefore, it is crucial to quantify the residual ratio  $\alpha$  so the model can be well-trained instead of underfitting or overfitting.

Recent works such as Tip-Adapter [274] and CLIP-Adapter [73] involve feature connection while they set such ratio as a fixed hyper-parameter that needs to be tuned during the training process. We argue the residual ratio should be dynamic and self-adaptive during training instead of fixed, as the learning status changes along the training process. The learning status is the key to measuring the model’s fit for the new tasks. Besides, when we have multiple downstream tasks, it is time-consuming and computation-inefficient to manually tune the hyper-parameter for every single task. Hence we propose a self-adaptive residual ratio in this subsection.

**DVC: Dual-view Vision Contrastive.** Inspired by self-supervised learning (e.g., SimCLR [37]), the composition of data augmentations is relevant to define effective predictive tasks. Therefore, we propose Dual-view Vision Contrastive (DVC) to measure the model’s learning status. Given an image  $x$ , we leverage two types of augmentations: “weak” and “strong” [17]. “weak” refers to the standard flip-and-shift augmentation strategy while “strong” refers to perturb the appearance (RandAugment [202] and CTAugment [30]). We propose to measure the distance between representations from both augmentations, e.g., cosine similarity has shown good performance in both visual contrastive [37] and textual contrastive [290] tasks.

With the input  $x$ , we will get  $w(x)$  from the weak augmentation and  $s(x)$  from the strong augmentation. We obtain  $F(w(x))$  and  $F(s(x))$  after the vision encoder  $F$ . We define the distance as  $\tau$ :

$$\tau = \frac{1}{N} \sum_{i=1}^N \cos(F(w(x_i)), F(s(x_i))), \quad (8.6)$$

where  $N$  is the number of samples in the training set, and  $\cos$  is a cosine similarity. The model’s

learning status is relevant to its consistency across different perturbations. We traverse the entire training set and calculate its mean value to represent the model’s performance.

**Bound constraints.** In addition to DVC that measures the predictive consistency of the model, we also select “accuracy” on the training set as a second metric to measure the learning status, denoted as  $T$ . Empirically, “accuracy” alone cannot fully tell if the model is well-fit: 100% of accuracy may indicate overfitting, which is the biggest challenge when we adapt a large model to a downstream task where the training set is very small. However, “accuracy” is a good metric to show if the model is underfitting. Therefore, we propose to fuse DVC and “accuracy” to represent the model’s status during training.

If we look at  $\alpha$  and set it fixed, we can find the model is prone to overfit if  $\alpha$  is too small as it will heavily rely on the fine-tuned knowledge; the model tends to underfit if  $\alpha$  is too large as it prevents the model from learning new knowledge (directly use the original pre-trained knowledge). Therefore, we propose a dynamic way with lower and upper bounds to obtain  $\alpha$  during training:

$$\alpha = \min(\max(\tau, T), 1 - \tau). \quad (8.7)$$

The bounds are associated with predictive consistency (DVC) and predictive accuracy on the training set. For example, if the model is overfitting,  $T$  will be large and  $\tau$  will be relatively small. Then  $\alpha$  will be smaller in the next epoch to use less fine-tuned features. Therefore, our proposed bound constraints can effectively adjust  $\alpha$  to alleviate the overfitting issue.

### 8.3.4 Path-CLIP: end-to-end adaptation of CLIP to pathology tasks

To efficiently adapt CLIP to pathology image tasks, we design a residual feature connection with a self-adaptive residual ratio to balance the original and fine-tuned knowledge based on the learning status. To alleviate the overfitting issue, we incorporate Hidden Representation Perturbation and design bound constraints for the residual ratio. Besides these, to distill more visual representations from the new image domain, we add a contrastive loss to the loss function so that the trainable portion has a better classifying ability. We follow SimCLR [37] to define the

Contrastive Loss (CL) for one image in the batch as:

$$l_i^c = \begin{cases} -\log \frac{\exp(\text{sim}(v_i, v_{i+B})/\gamma)}{\sum_{k \neq i}^{2B} \exp(\text{sim}(v_i, v_k)/\gamma)} & , \text{ if } i < B, \\ -\log \frac{\exp(\text{sim}(v_i, v_{B-i})/\gamma)}{\sum_{k \neq i}^{2B} \exp(\text{sim}(v_i, v_k)/\gamma)} & , \text{ otherwise.} \end{cases} \quad (8.8)$$

Differing from SimCLR [37], as shown in Fig. 8.4, here we set image  $i$  as the weakly augmented image and image  $i + B$  as the strongly augmented image instead of two randomly augmented images as the positive pair. We named this loss as  $\mathcal{L}_{\text{SWAPP}}$ . **SWAPP** here refers to **Strong-Weak Augmented Positive Pair**.

Finally, the total loss for fine-tuning CLIP on our pathology datasets is as follows:

$$\begin{aligned} \mathcal{L}_{\text{total}} &= \mathcal{L}_{\text{CLIP}} + \mathcal{L}_{\text{SWAPP}}. \\ \mathcal{L}_{\text{CLIP}} &\rightarrow \frac{1}{2B} \sum_{i=1}^B (l_i^{v \rightarrow t} + l_i^{t \rightarrow v}), \\ \mathcal{L}_{\text{SWAPP}} &\rightarrow \frac{1}{2B} \sum_{i=1}^{2B} l_i^c \end{aligned} \quad (8.9)$$

We define this loss as **Doublet Multimodal Contrastive Loss (DMCL)** because the items involve both inter-contrastive (image and text) and intra-contrastive (image and image).

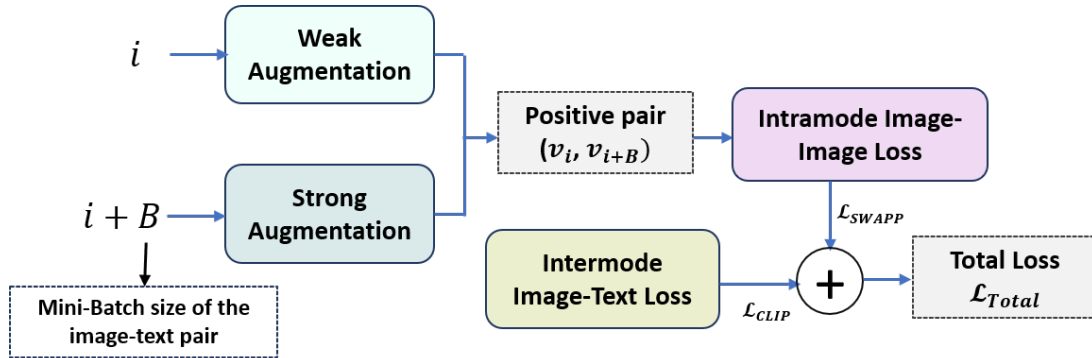


Figure 8.4. Doublet Multimodal Contrastive Loss (DMCL) that utilizes the traditional image-text pair and the proposed image-image pair with the strong and weak augmentation criteria.

## 8.4 Experiments

### 8.4.1 Datasets and setup

**Pathology image classification tasks.** The datasets used to validate our framework are collected from two distinct pathology projects [221, 222]. Both projects make their data available as

Table 8.2: Quantitative comparison on PCam. Precision, Recall, and F1-score refer to the macro-averaged values from all classes.

| Learning Manner | Labeled Ratio | Algorithm          | Accuracy                          | Precision                         | Recall                            | F1-score                          | AUC                               |
|-----------------|---------------|--------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| Supervised      | 100%          | -                  | 92.8 $\pm$ 0.30                   | 92.9 $\pm$ 0.20                   | 92.6 $\pm$ 0.21                   | 92.8 $\pm$ 0.20                   | 0.95 $\pm$ 0.01                   |
| Supervised      | 0.5%          | -                  | 45.1 $\pm$ 2.27                   | 42.0 $\pm$ 2.14                   | 33.6 $\pm$ 3.30                   | 37.3 $\pm$ 2.86                   | 0.51 $\pm$ 0.01                   |
| Self-supervised | 0.5%          | SimCLR [37]        | 60.4 $\pm$ 1.45                   | 63.3 $\pm$ 1.90                   | 58.5 $\pm$ 2.01                   | 60.8 $\pm$ 1.20                   | 0.61 $\pm$ 0.04                   |
| Semi-supervised | 0.5%          | Pseudo-Label [147] | 55.7 $\pm$ 2.05                   | 59.2 $\pm$ 2.23                   | 54.5 $\pm$ 1.97                   | 56.8 $\pm$ 1.55                   | 0.58 $\pm$ 0.03                   |
|                 |               | FixMatch [17]      | 73.2 $\pm$ 0.76                   | 77.5 $\pm$ 0.50                   | 73.7 $\pm$ 0.25                   | 75.6 $\pm$ 0.37                   | 0.84 $\pm$ 0.04                   |
|                 |               | Dash [212]         | 71.0 $\pm$ 0.98                   | 75.3 $\pm$ 0.78                   | 70.2 $\pm$ 0.45                   | 72.7 $\pm$ 0.50                   | 0.82 $\pm$ 0.02                   |
|                 |               | FlexMatch [18]     | 74.0 $\pm$ 0.80                   | 78.1 $\pm$ 1.15                   | 73.0 $\pm$ 0.90                   | 75.5 $\pm$ 0.84                   | 0.85 $\pm$ 0.03                   |
| Multimodal CLIP | -             | Zero-shot [59]     | 56.5 $\pm$ 0.00                   | 57.4 $\pm$ 0.00                   | 50.3 $\pm$ 0.00                   | 53.7 $\pm$ 0.00                   | 0.60 $\pm$ 0.00                   |
|                 | 0.5%          | CoOp [241]         | 63.6 $\pm$ 0.25                   | 63.9 $\pm$ 0.42                   | 62.5 $\pm$ 0.35                   | 63.0 $\pm$ 0.29                   | 0.67 $\pm$ 0.02                   |
|                 |               | CLIP-Adapter [73]  | 72.3 $\pm$ 1.02                   | 77.2 $\pm$ 0.95                   | 63.2 $\pm$ 0.56                   | 69.4 $\pm$ 0.84                   | 0.81 $\pm$ 0.02                   |
|                 |               | TIP-Adapter [274]  | 73.9 $\pm$ 0.50                   | 70.5 $\pm$ 0.85                   | 78.0 $\pm$ 0.49                   | 74.0 $\pm$ 0.95                   | 0.84 $\pm$ 0.02                   |
|                 |               | <b>Path-CLIP</b>   | <b>81.9 <math>\pm</math> 0.78</b> | <b>79.4 <math>\pm</math> 0.35</b> | <b>85.0 <math>\pm</math> 0.80</b> | <b>82.1 <math>\pm</math> 0.95</b> | <b>0.89 <math>\pm</math> 0.02</b> |
| Supervised      | 1.5%          | -                  | 62.8 $\pm$ 1.31                   | 69.5 $\pm$ 0.67                   | 54.1 $\pm$ 2.27                   | 60.8 $\pm$ 1.68                   | 0.63 $\pm$ 0.02                   |
| Self-supervised | 1.5%          | SimCLR [37]        | 64.9 $\pm$ 1.02                   | 65.2 $\pm$ 1.45                   | 60.1 $\pm$ 1.70                   | 62.5 $\pm$ 1.05                   | 0.64 $\pm$ 0.03                   |
| Semi-supervised | 1.5%          | Pseudo-Label [147] | 67.9 $\pm$ 1.85                   | 70.5 $\pm$ 1.90                   | 62.2 $\pm$ 1.76                   | 66.1 $\pm$ 1.42                   | 0.67 $\pm$ 0.04                   |
|                 |               | FixMatch [17]      | 83.9 $\pm$ 1.29                   | 85.1 $\pm$ 0.85                   | 84.1 $\pm$ 0.95                   | 84.6 $\pm$ 0.90                   | 0.87 $\pm$ 0.01                   |
|                 |               | Dash [212]         | 82.8 $\pm$ 1.09                   | 83.0 $\pm$ 0.55                   | 81.2 $\pm$ 0.60                   | 82.1 $\pm$ 0.78                   | 0.85 $\pm$ 0.03                   |
|                 |               | FlexMatch [18]     | 84.1 $\pm$ 1.02                   | 84.9 $\pm$ 0.75                   | 84.4 $\pm$ 0.80                   | 84.6 $\pm$ 0.75                   | 0.86 $\pm$ 0.04                   |
| Multimodal CLIP | 1.5%          | CoOp [241]         | 61.5 $\pm$ 0.75                   | 60.5 $\pm$ 1.03                   | 65.6 $\pm$ 0.90                   | 63.1 $\pm$ 0.85                   | 0.65 $\pm$ 0.02                   |
|                 |               | CLIP-Adapter [73]  | 74.1 $\pm$ 0.88                   | 79.1 $\pm$ 0.34                   | 65.4 $\pm$ 0.89                   | 71.6 $\pm$ 1.02                   | 0.82 $\pm$ 0.01                   |
|                 |               | TIP-Adapter [274]  | 75.2 $\pm$ 0.94                   | 72.3 $\pm$ 0.69                   | 79.4 $\pm$ 0.86                   | 75.7 $\pm$ 0.73                   | 0.83 $\pm$ 0.02                   |
|                 |               | <b>Path-CLIP</b>   | <b>83.1 <math>\pm</math> 0.67</b> | <b>82.8 <math>\pm</math> 0.95</b> | <b>82.0 <math>\pm</math> 0.75</b> | <b>82.4 <math>\pm</math> 0.46</b> | <b>0.91 <math>\pm</math> 0.02</b> |

patches, which are extracted from H&E (Hematoxylin and Eosin) stained Whole Slide Images (WSI) digitized from Formalin-Fixed Paraffin-Embedded (FFPE) slides. Each dataset aims to detect different cancerous tissue.

**PatchCamelyon (PCam) [221].** PCam is a patch dataset extracted from Camelyon16 challenge [291]. The original 400 WSIs from Camelyon16 were digitized breast tissue with potential metastasized cancerous tissue on the lymph nodes. The original WSIs were scanned at  $40\times$  resolution but later downsampled to  $10\times$ . The WSIs were collected from two different centers. PCam extracted 327,680 patches at  $96\times 96$  resolution and labeled them positive or negative. Positive labeled patches present tumor tissue in the central  $32\times 32$  patch region. There are an equal amount of positive and negative labeled samples.

**MHIST [222].** MHIST contains 3,152 patches from colorectal regions at  $224 \times 224$  pixel resolution. These patches were extracted from 328 WSIs scanned at  $40\times$  resolution. Each patch may be labeled Hyperplastic Polyp (HP) or Sessile Serrated Adenoma (SSA). HP is the majority class with 68.59% of labels. Labeling colorectal polyps between HP and SSA is challenging due to high inter-pathologist disagreement. Seven pathologists contributed to the ground truth to ensure reliable labels.

**GM/WM segmentation task [29].** Besides the two pathology classification tasks above, we also evaluate our proposed approach on a grey/white matter segmentation task in pathological slides. For this dataset, we have 30 gigapixel slides with an average resolution of  $50,000 \times 60,000$  by an Aperio AT2 scanner at 20X magnification. They were collected and annotated by a trained researcher and expert at the University of California Davis, Alzheimer’s Disease Research Center (UCD-ADRC). We follow the previous work [29] to split 20 slides into the training/validation set while 10 slides into the hold-out test set. We tile these gigapixel slides into patches at  $256 \times 256$  and convert it into a weakly-supervised task [292].

**Experimental setup.** For a fair comparison, we use ResNet-50 [55] as the backbone in the vision encoder  $f(\cdot)$  and BERT [279] as the textual encoder. We follow CLIP [59] to use gradient scaling to facilitate mixed-precision training. We set the learning rate as 0.0001. The batch size is 32. We use Adam[220] as the optimizer. To compare the computational complexity, we conduct all experiments on a single GPU (Nvidia RTX 2080Ti).



Table 8.3: Quantitative comparison on MHIST. Precision, Recall, and F1-score refer to the macro-averaged values from all classes.

| Learning Manner | Labeled Ratio | Algorithm          | Accuracy                          | Precision                         | Recall                            | F1-score                          | AUC                               |
|-----------------|---------------|--------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| Supervised      | 100%          | -                  | 86.0 $\pm$ 0.36                   | 84.8 $\pm$ 0.61                   | 83.5 $\pm$ 0.55                   | 84.1 $\pm$ 0.58                   | 0.89 $\pm$ 0.01                   |
| Self-supervised | 10%           | SimCLR [37]        | 63.2 $\pm$ 0.95                   | 63.8 $\pm$ 1.20                   | 62.2 $\pm$ 1.70                   | 63.0 $\pm$ 0.85                   | 0.66 $\pm$ 0.04                   |
| Semi-supervised | 10%           | Pseudo-Label [147] | 65.4 $\pm$ 1.04                   | 66.5 $\pm$ 0.98                   | 65.9 $\pm$ 0.85                   | 66.2 $\pm$ 1.42                   | 0.75 $\pm$ 0.03                   |
|                 |               | FixMatch [17]      | 70.5 $\pm$ 1.55                   | 68.8 $\pm$ 2.76                   | 69.5 $\pm$ 2.97                   | 69.1 $\pm$ 2.86                   | 0.79 $\pm$ 0.02                   |
|                 |               | Dash [212]         | 70.2 $\pm$ 1.21                   | 70.0 $\pm$ 0.85                   | 71.3 $\pm$ 0.69                   | 70.6 $\pm$ 1.48                   | 0.81 $\pm$ 0.04                   |
|                 |               | FlexMatch [18]     | 73.1 $\pm$ 0.84                   | 74.6 $\pm$ 1.05                   | 75.1 $\pm$ 0.90                   | 74.8 $\pm$ 0.89                   | 0.82 $\pm$ 0.03                   |
| Multimodal CLIP | 10%           | Zero-shot [59]     | 36.9 $\pm$ 0.00                   | 36.9 $\pm$ 0.00                   | 100 $\pm$ 0.00                    | 53.9 $\pm$ 0.00                   | 0.50 $\pm$ 0.00                   |
|                 |               | CoOp [241]         | 68.9 $\pm$ 0.40                   | 56.0 $\pm$ 0.65                   | 72.8 $\pm$ 0.53                   | 63.3 $\pm$ 0.86                   | 0.76 $\pm$ 0.01                   |
|                 |               | CLIP-Adapter [73]  | 65.9 $\pm$ 1.45                   | 52.2 $\pm$ 1.23                   | 87.2 $\pm$ 0.85                   | 65.3 $\pm$ 0.95                   | 0.77 $\pm$ 0.02                   |
|                 |               | TIP-Adapter [274]  | 63.1 $\pm$ 0.78                   | 49.9 $\pm$ 0.90                   | 94.7 $\pm$ 1.02                   | 65.4 $\pm$ 0.56                   | 0.78 $\pm$ 0.03                   |
|                 |               | <b>Path-CLIP</b>   | <b>74.8 <math>\pm</math> 1.95</b> | <b>63.3 <math>\pm</math> 1.40</b> | <b>75.6 <math>\pm</math> 0.98</b> | <b>68.9 <math>\pm</math> 1.50</b> | <b>0.84 <math>\pm</math> 0.02</b> |

**Baselines.** We first implement traditional supervised learning (SL) methods with the entire dataset to explore the upper-bound performance for each task. Then we reduce the data usage and investigate the performance change of these SL methods. Then we compare Path-CLIP with original CLIP [59] and three recent CLIP-based fine-tuning methods: CoOp [241], Tip-Adapter [274] and CLIP-Adapter [73]. We compare them under the limited data scenarios to test their ability to do downstream tasks when the new dataset is small. Besides that, we also incorporate recent self-supervised learning (SimCLR [37]) and semi-supervised learning (Pseudo-Label [147], FixMatch [17], Dash [212], and FlexMatch [18]). The semi-supervised training also requires a large amount of unlabeled data, which involves data collection efforts. We compare our proposed CLIP-based method with semi-supervised learning to show the advantages of reducing data collection efforts.

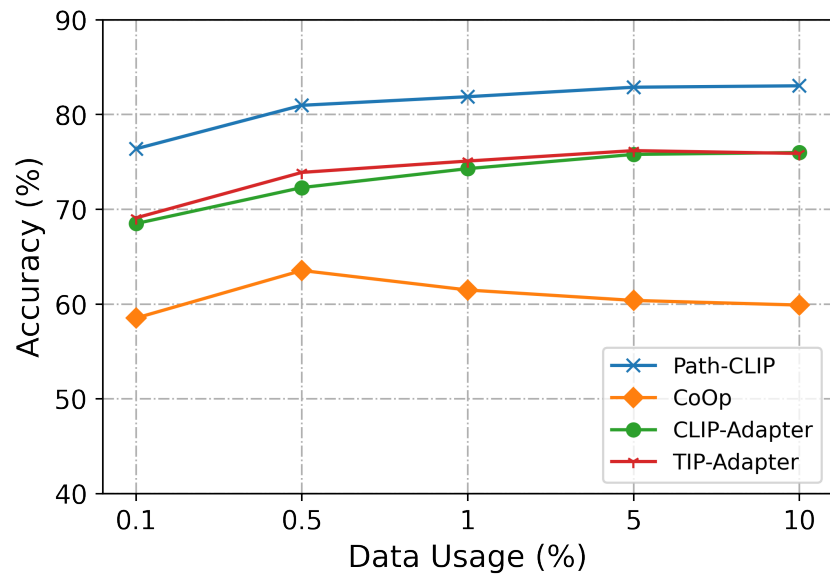
## 8.4.2 Main results

**Classification task: PCam [221] and MHIST [222].** The main results for two classification benchmarks are summarized in Table 8.2 and Table 8.3. First, CLIP shows its promising zero-shot ability: when we directly test CLIP on PCam [221], it can get 56.5% accuracy. Then, we compare Path-CLIP with other CLIP-based fine-tuning methods. In PCam [221], we present 0.5% and 1.5% data usage in training set to fine-tune CLIP, respectively. We find all of these

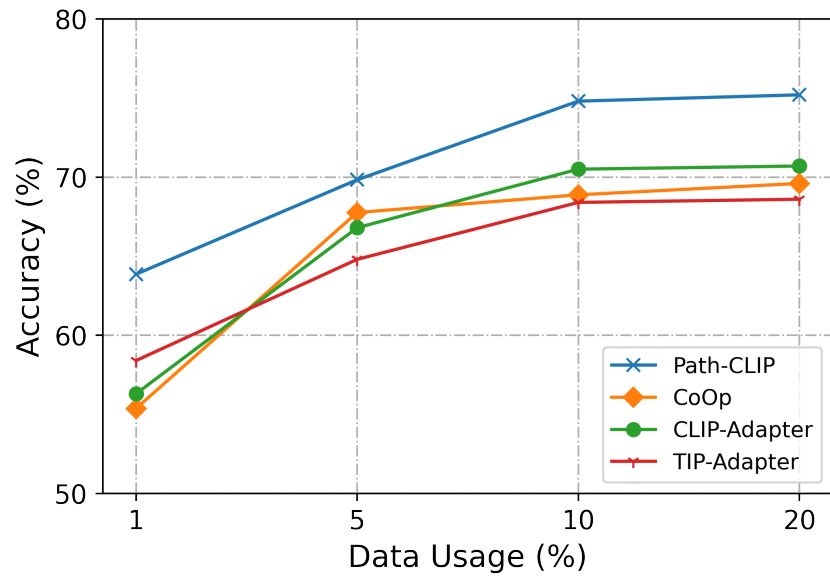
fine-tuning algorithms [241, 73, 274] can improve the performance of the original CLIP with the new data from the downstream task presented. Our proposed Path-CLIP can outperform them in both settings. Specifically, Path-CLIP outperforms TIP-Adapter [274] with 8% and 7.9% accuracy when only 0.5% and 1.5% data are used, respectively. Path-CLIP also shows similar improvement in MHIST [222] (Table 8.3). Since MHIST [222] is small, we use 10% of data for fine-tuning. On the other hand, we implemented several recent semi-supervised learning where the rest of the data in the training set are used as an unlabeled set. Although Path-CLIP does not use unlabeled data, it can achieve competitive results with semi-supervised learning. This indicates the potential of Path-CLIP to reduce the efforts of data collection.

Besides these ratios of data usage, we also visualize the results by varying the ratio to show the performance of each fine-tuning algorithm. As shown in Fig. 8.5, CoOp [241] may not bring improvement to PCam when more data are used to fine-tune the prompt. CoOp freezes the encoders and only fine-tunes the language prompt part, while other algorithms also fine-tune and vision part. We conclude that visual fine-tuning is more critical compared to language fine-tuning, which is also reported in [73]. In Fig. 8.5, we also show that our proposed CLIP-Path can consistently outperform the other three recent fine-tuning methods when we vary the data usage.

**Segmentation task: GM/WM segmentation [29].** Besides two classification tasks, we follow a recent work [29] to generalize our method to a weakly-supervised segmentation problem. We select IoU and DICE scores as the measuring metrics for the grey matter (GM) and white matter (WM) regions. The results are summarized in Table 8.4. We select two supervised learning (SL) approaches for the segmentation task: FCN [11] and U-Net [27]: we first use all data in the training set (100%) to train them in a supervised manner, then we reduce the dataset to 0.1% and 1% setting and train the model again. We can see that both SL algorithms face performance degradation when the training data is scarce. They require an extensive and well-curated labeled dataset to get satisfactory results. We then focus on multi-modal training and fine-tuning CLIP with recent algorithms. Table 8.4 shows Path-CLIP can achieve superior results among other fine-tuning techniques.



(a) PCam



(b) MHIST

Figure 8.5. Comparison of different fine-tuning methods by varying data usage for the adaptation.

Table 8.4: Performance comparison on GM/WM segmentation.

| Data Usage  | Algorithm         | GM_IoU       | GM_DICE      | WM_IoU       | WM_DICE      |
|-------------|-------------------|--------------|--------------|--------------|--------------|
| 100% (SL)   | FCN [11]          | 71.72        | 83.21        | 49.09        | 66.08        |
|             | U-Net [27]        | 91.52        | 95.17        | 82.02        | 88.15        |
| 0.1% (SL)   | FCN [11]          | 48.25        | 25.01        | 24.54        | 30.02        |
|             | U-Net [27]        | 46.13        | 50.10        | 5.75         | 12.23        |
| 0.1% (CLIP) | CoOp [241]        | 62.23        | 65.70        | 39.92        | 48.65        |
|             | CLIP-Adapter [73] | 67.80        | 74.32        | 43.25        | 54.61        |
|             | TIP-Adapter [274] | 69.92        | 75.50        | 46.23        | 56.60        |
|             | <b>Path-CLIP</b>  | <b>78.05</b> | <b>83.70</b> | <b>54.50</b> | <b>63.23</b> |
| 1% (SL)     | FCN [11]          | 51.34        | 30.20        | 31.02        | 35.12        |
|             | U-Net [27]        | 50.24        | 53.23        | 15.43        | 25.58        |
| 1% (CLIP)   | CoOp [241]        | 63.45        | 64.12        | 37.85        | 45.34        |
|             | CLIP-Adapter [73] | 69.99        | 76.54        | 47.95        | 58.34        |
|             | TIP-Adapter [274] | 72.12        | 77.30        | 55.65        | 63.60        |
|             | <b>Path-CLIP</b>  | <b>80.21</b> | <b>88.70</b> | <b>60.50</b> | <b>70.34</b> |

### 8.4.3 Ablation studies

#### 8.4.3.1 Vision vs prompt fine-tuning

CoOp [241] is the recent state-of-the-art fine-tuning method for CLIP. Hence, we mainly compare our proposed RFC with its performance and computational complexity. We summarize the results in Table 8.5. When we range the data usage from 0.1% to 1%, we reveal Path-CLIP can obtain 5.5% improvement in accuracy while only introducing 1 minute of additional training time. However, CoOp [241] has the overfitting issue and obtains a lower score while using 53 minutes, which is almost five times compared to the time used by RFC. The situation remains similar under the 10% of data usage. We conclude our proposed RFC can get over 25% improvement on CLIP by only using 10 minutes for the fine-tuning on a single GPU, which is promising for digital pathology research. Table 8.5 demonstrates results revealing vision fine-tuning for a

Table 8.5: Vision vs prompt fine-tuning on PCam.

| Data Usage | Algorithm        | Accuracy    | Training Time        |
|------------|------------------|-------------|----------------------|
| Zero-shot  | CLIP [59]        | 56.5        | -                    |
| 0.1%       | CoOp [241]       | 58.5        | <b>7 min 6 sec</b>   |
|            | <b>Path-CLIP</b> | <b>76.4</b> | 10 min 29 sec        |
| 1%         | CoOp [241]       | 61.5        | 53 min 21 sec        |
|            | <b>Path-CLIP</b> | <b>81.9</b> | <b>11 min 56 sec</b> |
| 10%        | CoOp [241]       | 59.9        | 2 h 23 min 45 sec    |
|            | <b>Path-CLIP</b> | <b>83.0</b> | <b>27 min 18 sec</b> |

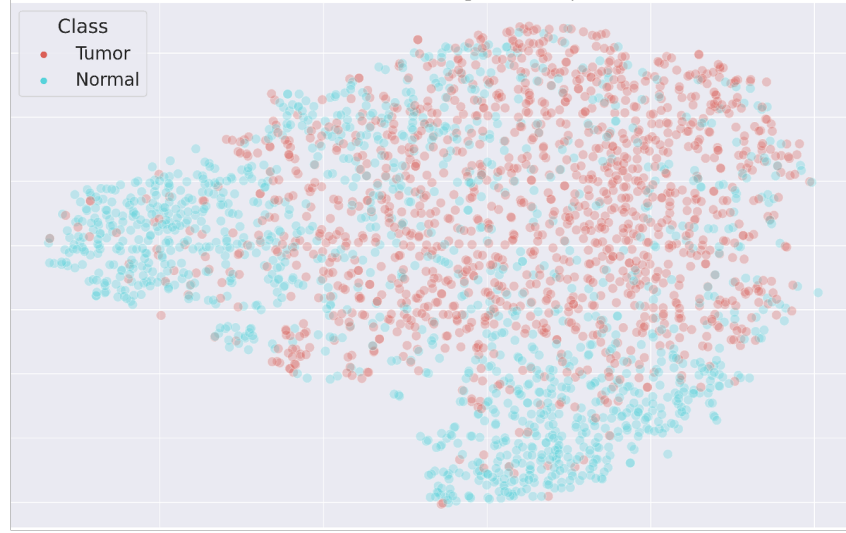
medical domain may be more effective than prompt fine-tuning.

#### 8.4.3.2 Effect of Residual Feature Connection

To show the effect of the Residual Feature Connection (RFC), we use t-SNE [293] to visualize the manifold of the original CLIP and CLIP with RFC after training it on PCam [221]. The t-SNE results are presented in Fig. 8.6: it illustrates RFC can show clear separations of image features belonging to different classes in the high-dimensional classification space. In addition, as shown in Table 8.6, RFC with the self-adaptive ratio can significantly improve with +15.5% accuracy.

#### 8.4.4 Importance of alleviating overfitting

Since the new downstream task may have a very limited training set, it is crucial to alleviate the potential overfitting issue. Besides the RFC with the self-adaptive ratio, we also propose HRP (Hidden Representation Perturbation), DVC (Dual-view Vision Contrastive), and DMCL (Doublet Multimodal Contrastive Loss) to cope with the overfitting issue. Table 8.6 summarizes the contribution of each proposed module: we find HRP can improve with +2.2% accuracy and DVC can improve with +4.3%. DMCL inspired by self-supervised learning can also aid in



(a) CLIP



(b) CLIP with RFC

Figure 8.6. Visualization of different learned feature manifolds via t-SNE.

great improvements. We conclude contrastive learning and perturbations on data and hidden representations are helpful to the overfitting issue when the new dataset is small.

## 8.5 Discussion

Automated analysis of the pathology image task is an important aspect in providing scalable objective evaluations of medical images. Limited data availability limits the intensive end-to-end

Table 8.6: Ablation study on PCam using 0.5% data for fine-tuning.

| # | RFC | HRP | DVC | DMCL | Accuracy | F1-score | AUC  |
|---|-----|-----|-----|------|----------|----------|------|
| 1 | ✗   | ✗   | ✗   | ✗    | 56.5     | 53.7     | 0.60 |
| 2 | ✓   | ✗   | ✗   | ✗    | 72.0     | 70.5     | 0.69 |
| 3 | ✓   | ✓   | ✗   | ✗    | 74.2     | 75.5     | 0.78 |
| 4 | ✓   | ✓   | ✓   | ✗    | 78.5     | 79.3     | 0.84 |
| 5 | ✓   | ✓   | ✓   | ✓    | 81.9     | 82.1     | 0.89 |

training of deep networks. Also, the requirement of multiple learning tasks for the complete pathology image analysis is time-consuming and can be resource-hungry. In this work, we explore the generalization of CLIP in pathology image classification. The Contrastive Language-Image Pre-training (CLIP) model has shown powerful capabilities in diverse downstream applications. However, its applicability in pathology image analysis with limited labeled data is still under study due to the giant domain shift and overfitting issues. We propose Path-CLIP to efficiently fine-tune CLIP image encoder that extracts the rich semantic information using small datasets and light computing resources.

We introduce the self-adaptive residual ratio  $\alpha$  into Residual Feature Connection (RFC), which changes dynamically during training and overcomes the overfitting issue. The ablation studies show the importance of self-adaptive residual ratio, an improvement of 15.5% and 16.8% is observed in accuracy and F1 score, respectively. Additionally, other modules of the image encoder of the vision-language pre-trained model significantly contribute to improving the prediction ability of the model and address the overfitting issue as well. The combination of these modules resulting in an overall improvement of 9.9% in accuracy over RFC. The results reveal the superiority and effectiveness of the proposed Path-CLIP in comparison to the existing fine-tuning methods, with respect to the evaluation criteria considered. It is interesting to note the Path-CLIP outperforms the traditional CLIP by 19.9% when using only 0.1% of the labeled data in PCam dataset, in addition, the less computational time compliments the efficiency of

the method. Hence, we conclude Path-CLIP is more robust and has the potential to bridge the domain shift between the pre-trained natural images and pathology images in an efficient and cost-effective manner.



# Chapter 9

## Future Work

### 9.1 Heterogeneity in pathology image analysis tasks

#### 9.1.1 Expanded datasets for robustness analysis

In Chapter 2 and 3, we only evaluated the proposed methods on Amyloid- $\beta$  stained WSIs from the temporal cortex neuroanatomical region. To extend the versatility of our approaches, we will incorporate WSIs from other neuroanatomical regions of the human brain for evaluation, such as superior middle temporal gyrus, middle frontal gyrus, and superior parietal lobule. Besides Amyloid- $\beta$  staining, we plan to evaluate on other staining WSIs, such as Hematoxylin and Eosin (H&E) staining. Figure 9.1 is one example of the evaluation on region parietal. We also plan to optimize the entire pipeline to reduce the computational cost and make it more user-friendly.

#### 9.1.2 Extend SSL to more pathology image analysis tasks

Pathology image analysis includes classification, segmentation, and object detection. All of them heavily rely on a well-curated and fully annotated dataset when training a supervised learning model. To minimize the labeling efforts, we propose to evaluate the proposed algorithms on diverse pathology image tasks, such as plaque detection [16] to further evaluate the effectiveness of the proposed methods.

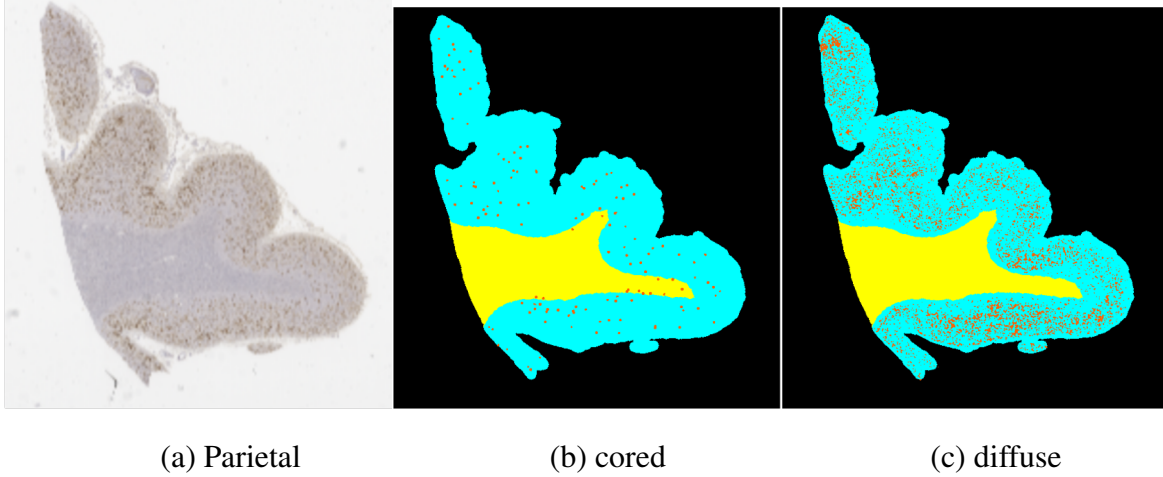


Figure 9.1. One example from region parietal. (b) and (c) shows the distribution of cored and diffuse in GM and WM, respectively.

### 9.1.3 Class-imbalanced and fairness issue in pathology images

Another common issue, data imbalance can also prevent the direct adaption of state-of-the-art SSL on our pathology images. As we have proposed SAW (in Chapter 4) that never makes any assumption on the unlabeled data, it is promising to apply the enhanced SSL framework with SAW to pathology image analysis tasks and achieve well-defined performance with scarce labeled data. Another challenge of deep learning is the model’s bias and fairness issue: we plan to propose an optimal trade-off between fairness and accuracy for any deep learning models with bi-level optimization algorithms.

## 9.2 Transferability and semi-supervised transfer learning

**Transferability.** Although we have proved the applicability of semi-supervised learning on some applications, it still requires a large amount of unlabeled data. As the data extraction process for pathology images is expensive and less frequent [294], it is expected that the number of labeled medical images can be limited, which sets an obstacle for applying SSL. Transfer learning can be an effective way when only a few labeled target images are available for supervision [295, 296]. However, the transferability of pre-trained models to pathology image analysis is still under exploration [294], especially the transferability measurement. A transferability metric aims to quantitatively reveal how feasible it is to transfer knowledge learned from a source task to a

target task [297, 298, 299, 300, 301]. Although several theoretical results in transfer learning such as [302, 303, 304] indicate that an inevitable generating distribution divergence between the source and target data can model the task relationship, this relationship is difficult to estimate in practice when target data is limited. Previous transferability metrics [305, 306] rely on training loss or validation accuracy to calculate the task relationships. Thus they require to retrain the source model, which may involve expensive computation. Recent analytical metrics [69, 307, 66] have strict assumptions on data: NCE [308] assumes that the target tasks share the same input instances as the source tasks; H-score [307] makes an assumption that the target data are distributed in the same domain as the source data.

A recent study [301] proposed a novel cross-domain cross-task transferability metric to learn the domain difference and task difference in a unified framework. This approach is about 75x faster compared to traditional transfer learning, which is promising for applications on gigapixel pathology images. Hence such a metric has the potential to boost the applicability of transfer learning in pathology images by selecting an appropriately transferable way to achieve competitive results even when available data are very limited.

**Semi-supervised transfer learning.** Although semi-supervised learning algorithms have shown their promising results in leveraging both labeled and unlabeled data, most of them assume a general setting where the model is trained with randomly initialized parameters from scratch [309, 310]. In [310], they consider a more realistic setting by utilizing a pre-trained model that is powerful and effectively fits on large-scale datasets for general purposes. These models have been empirically proven to keep good transferability on various down-streaming tasks [305]. They can achieve significant improvement on the generalization capacity of target tasks, especially when data are relatively scarce. However, the efforts on the exploration of a systematic solution on semi-supervised transfer learning remain limited. Abuduweili *et al.* [310] proposed a semi-supervised transfer learning algorithm beyond a simple combination of these two learning schemes. They borrow the compelling idea of consistency loss in semi-supervised learning and adapt it into inductive transfer learning.

### 9.3 From interpretability to continual learning

In pathology image settings, we have WSIs with different staining and from different cases. Training a recursive model with the process of data collection requires prior knowledge preserve and new knowledge assimilation [311]. Continual learning, also known as incremental learning or life-long learning, can assimilate new information from new tasks without forgetting past knowledge learned from previous data and tasks [312, 313, 314]. It has the potential to achieve such training in pathology image analysis. Traditional continual learning can be roughly partitioned into three groups: 1) regularization-based methods that may rely on knowledge distillation [315, 316, 317]; 2) exemplar-based methods that add partial copies of previous data into the current training set [318, 319, 313]; 3) architecture-based methods that attempt to modify some components in the architecture for fitting in the task [320, 321]. However, in practice, these methods suffer from inconsistent performance across different datasets, and their ability to mitigate the **catastrophic forgetting** [322] is still questionable [323].

Recently in the interpretability research field, Chan *et al.* [324] proposed ReDuNet and claimed it is a “white-box” (purely interpretable) model that does not need any propagation for each layer. Furthermore, they proved that the iterative gradient ascent scheme for optimizing the rate reduction objective could naturally lead to building a model with the same structure of a multi-layer deep network. With such a “white-box” model, they designed the architecture derived from the principle of rate reduction. They claimed that the model built this way can preserve knowledge by adequately adjusting the model parameters to preserve knowledge learned from previous training data and tasks, resulting in the feasibility of continual learning naturally [323]. Therefore, studying the interpretability of deep learning models can be a potential direction to build a robust and practical continual learning model under real-world settings. On the other hand, the model’s interpretability is also a critical factor in medical image analysis by providing meaningful medical features to doctors. The exploration of developing a “white-box” deep learning model for pathology image tasks is under study. Hence we propose further to study the interpretability of networks [26] to aid in medical research.

## REFERENCES

- [1] T. Rutherford and A. Socio, "Population ageing: statistics," *House of Commons library (Standard not. Retrieved Jan 2, 2013, from: www.parliament.uk/topics/PopulationArchive*, 2012.
- [2] A. Association *et al.*, "2020 Alzheimer's Disease facts and figures," *Alzheimer's & Dementia*, vol. 16, no. 3, pp. 391–460, 2020.
- [3] B. N. Dugger and D. W. Dickson, "Pathology of neurodegenerative diseases," *Cold Spring Harbor perspectives in biology*, vol. 9, no. 7, p. a028035, 2017.
- [4] S. Al-Janabi, A. Huisman, and P. J. Van Diest, "Digital pathology: current status and future perspectives," *Histopathology*, vol. 61, no. 1, pp. 1–9, 2012.
- [5] P. Bankhead and M. B. Loughrey *et al.*, "QuPath: Open source software for digital pathology image analysis," *Scientific Reports*, Dec 2017.
- [6] C. Higgins, "Applications and challenges of digital pathology and whole slide imaging," *Biotech. Histochem.*, vol. 90, no. 5, pp. 341–347, 2015.
- [7] L. Pantanowitz and P. N. t. Valenstein, "Review of the current state of whole slide imaging in pathology," *J. Pathol. Inform.*, vol. 2, 2011.
- [8] B. N. Dugger, M. Tu, M. E. Murray, and D. W. Dickson, "Disease specificity and pathologic progression of tau pathology in brainstem nuclei of alzheimer's disease and progressive supranuclear palsy," *Neuroscience Letters*, vol. 491, no. 2, pp. 122–126, 2011.
- [9] J. H. Neltner, E. L. Abner, and F. A. Schmitt, "Digital pathology and image analysis for robust high-throughput quantitative assessment of Alzheimer disease neuropathologic changes," *J. Neuropathol. Exp. Neurol.*, vol. 71, no. 12, pp. 1075–1085, 2012.
- [10] N. Iwamoto, E. Nishiyama, J. Ohwada, and H. Arai, "Distribution of amyloid deposits in the cerebral white matter of the alzheimer's disease brain: relationship to blood vessels," *Acta neuropathologica*, vol. 93, no. 4, pp. 334–340, 1997.
- [11] P. Bándi, R. van de Loo, M. Intezar, D. Geijs, F. Ciompi, B. van Ginneken, J. van der Laak, and G. Litjens, "Comparison of different methods for tissue segmentation in histopathological whole-slide images," in *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*. IEEE, 2017, pp. 591–595.
- [12] D. Bug, F. Feuerhake, and D. Merhof, "Foreground extraction for histopathological whole slide imaging," in *Bildverarbeitung für die Medizin 2015*. Springer, 2015, pp. 419–424.
- [13] A. de Brebisson and G. Montana, "Deep neural networks for anatomical brain segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2015, pp. 20–28.

- [14] W. Chen, Z. Jiang, Z. Wang, K. Cui, and X. Qian, “Collaborative global-local networks for memory-efficient segmentation of ultra-high resolution images,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8924–8933.
- [15] Z. Lai, R. Guo, W. Xu, Z. Hu, K. Mifflin, B. N. Dugger, C.-N. Chuah, and S.-c. Cheung, “Automated grey and white matter segmentation in digitized  $\alpha\beta$  human brain tissue slide images,” in *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2020, pp. 1–6.
- [16] Z. Tang, K. V. Chuang, C. DeCarli, L.-W. Jin, L. Beckett, M. J. Keiser, and B. N. Dugger, “Interpretable classification of alzheimer’s disease pathologies with a convolutional neural network pipeline,” *Nature communications*, vol. 10, no. 1, pp. 1–14, 2019.
- [17] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, “Fixmatch: Simplifying semi-supervised learning with consistency and confidence,” in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 596–608.
- [18] B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, and T. Shinozaki, “Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling,” *NeurIPS*, vol. 34, pp. 18 408–18 419, 2021.
- [19] H. Chen, R. Tao, Y. Fan, Y. Wang, J. Wang, B. Schiele, X. Xie, B. Raj, and M. Savvides, “Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning,” *ICLR*, 2023.
- [20] Y. Wang, H. Chen, Q. Heng, W. Hou, M. Savvides, T. Shinozaki, B. Raj, Z. Wu, and J. Wang, “Freematch: Self-adaptive thresholding for semi-supervised learning,” *ICLR*, 2023.
- [21] J. V. Pulido, S. Guleria, L. Ehsan, M. Fasullo, R. Lippman, P. Mutha, T. Shah, S. Syed, and D. E. Brown, “Semi-supervised classification of noisy, gigapixel histology images,” in *2020 IEEE 20th International Conference on Bioinformatics and Bioengineering (BIBE)*. IEEE, 2020, pp. 563–568.
- [22] L. Chan, M. S. Hosseini, and K. N. Plataniotis, “A comprehensive analysis of weakly-supervised semantic segmentation in different image domains,” *International Journal of Computer Vision*, vol. 129, no. 2, pp. 361–384, 2021.
- [23] C. You, W. Dai, Y. Min, L. Staib, and J. S. Duncan, “Bootstrapping semi-supervised medical image segmentation with anatomical-aware contrastive distillation,” *arXiv preprint arXiv:2206.02307*, 2022.
- [24] C. You, W. Dai, Y. Min, L. Staib, J. S. Sekhon, and J. S. Duncan, “Action++: Improving semi-supervised medical image segmentation with adaptive anatomical contrast,” *arXiv preprint arXiv:2304.02689*, 2023.

- [25] L. C. Oliveira, Z. Lai, D. Harvey, K. Nzenkue, L.-W. Jin, C. Decarli, C.-N. Chuah, and B. N. Dugger, “Preanalytic variable effects on segmentation and quantification machine learning algorithms for amyloid- $\beta$  analyses on digitized human brain slides,” *Journal of Neuropathology & Experimental Neurology*, vol. 82, no. 3, pp. 212–220, 2023.
- [26] Z. Lai, R. Guo, W. Xu, Z. Hu, K. Mifflin, C. DeCarli, B. N. Dugger, S.-c. Cheung, and C.-N. Chuah, “Automated segmentation of amyloid- $\beta$  stained whole slide images of brain tissue,” *bioRxiv 2020.11.13.381871*, 2020.
- [27] K. Oskal, M. Risdal, E. Janssen, E. Undersrud, and T. Gulsrud, “A U-net based approach to epidermal tissue segmentation in whole slide histopathological images,” *SN Applied Sciences*, vol. 1, p. 672, 06 2019.
- [28] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *IEEE ICCV*, 2017, pp. 618–626.
- [29] Z. Lai, L. C. Oliveira, R. Guo, W. Xu, Z. Hu, K. Mifflin, C. Decarli, S.-C. Cheung, C.-N. Chuah, and B. N. Dugger, “Brainsec: Automated brain tissue segmentation pipeline for scalable neuropathological analysis,” *IEEE Access*, vol. 10, pp. 49 064–49 079, 2022.
- [30] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, “Mix-match: A holistic approach to semi-supervised learning,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
- [31] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. Goodfellow, “Realistic evaluation of deep semi-supervised learning algorithms,” *Advances in Neural Information Processing Systems*, vol. 31, pp. 3235–3246, 2018.
- [32] J. Kim, Y. Hur, S. Park, E. Yang, S. J. Hwang, and J. Shin, “Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 14 567–14 579.
- [33] C. Wei, K. Sohn, C. Mellina, A. Yuille, and F. Yang, “Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning,” *arXiv preprint arXiv:2102.09559*, 2021.
- [34] M. N. Rizve, K. Duarte, Y. S. Rawat, and M. Shah, “In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning,” *arXiv:2101.06329*, 2021.
- [35] K. S. Tai, P. Bailis, and G. Valiant, “Sinkhorn label allocation: Semi-supervised classification via annealed self-training,” *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.
- [36] Z. Lai, C. Wang, Z. Hu, B. N. Dugger, S.-C. Cheung, and C.-N. Chuah, “A semi-supervised learning for segmentation of gigapixel histopathology images from brain tissues,” in *2021*

- [37] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.
- [38] M. A. Rahman and Y. Wang, “Optimizing intersection-over-union in deep neural networks for image segmentation,” in *International symposium on visual computing*. Springer, 2016, pp. 234–244.
- [39] Z. Lai, C. Wang, H. Gunawan, S.-C. Cheung, and C.-N. Chuah, “Smoothed adaptive weighting for class-imbalanced semi-supervised learning,” in *NeurIPS (Under Review)*, 2021.
- [40] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel, “Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring,” in *International Conference on Learning Representations (ICLR)*, 2020.
- [41] Z. Lai, C. Wang, H. Gunawan, S.-C. S. Cheung, and C.-N. Chuah, “Smoothed adaptive weighting for imbalanced semi-supervised learning: Improve reliability against unknown distribution data,” in *ICML*, 2022, pp. 11 828–11 843.
- [42] Z. Lai, C. Wang, S.-c. Cheung, and C.-N. Chuah, “Sar: Self-adaptive refinement on pseudo labels for multiclass-imbalanced semi-supervised learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4091–4100.
- [43] M.-F. Balcan, A. Beygelzimer, and J. Langford, “Agnostic active learning,” *Journal of Computer and System Sciences*, vol. 75, no. 1, pp. 78–89, 2009.
- [44] K. Konyushkova, R. Sznitman, and P. Fua, “Learning active learning from data,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [45] Z. Lai, C. Wang, L. Cerny Oliveira, B. N. Dugger, S.-C. Cheung, and C.-N. Chuah, “Joint semi-supervised and active learning for segmentation of gigapixel pathology images with cost-effective labeling,” in *2021 IEEE International Conference on Computer Vision (ICCV) Workshop on Computational Challenges in Digital Pathology*, 2021.
- [46] W. R. Crum, O. Camara, and D. L. Hill, “Generalized overlap measures for evaluation and validation in medical image analysis,” *IEEE transactions on medical imaging*, vol. 25, no. 11, pp. 1451–1461, 2006.
- [47] Y. Ganin and V. Lempitsky, “Unsupervised domain adaptation by backpropagation,” in *ICML*, 2015, pp. 1180–1189.
- [48] T. Sun, C. Lu, T. Zhang, and H. Ling, “Safe self-refinement for transformer-based domain adaptation,” in *CVPR*, 2022, pp. 7191–7200.



- [49] T. Xu, W. Chen, P. Wang, F. Wang, H. Li, and R. Jin, “Cdtrans: Cross-domain transformer for unsupervised domain adaptation,” in *ICLR*, 2022.
- [50] Y. Zhang, T. Liu, M. Long, and M. I. Jordan, “Bridging theory and algorithm for domain adaptation,” in *ICML*, 2019.
- [51] J. Liang, D. Hu, and J. Feng, “Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation,” in *ICML*, 2020, pp. 6028–6039.
- [52] Z. Gao, S. Zhang, K. Huang, Q. Wang, and C. Zhong, “Gradient distribution alignment certificates better adversarial domain adaptation,” in *ICCV*, 2021, pp. 8937–8946.
- [53] S. Li, M. Xie, F. Lv, C. H. Liu, J. Liang, C. Qin, and W. Li, “Semantic concentration for domain adaptation,” in *ICCV*, 2021, pp. 9102–9111.
- [54] L. Chen, H. Chen, Z. Wei, X. Jin, X. Tan, Y. Jin, and E. Chen, “Reusing the task-specific classifier as a discriminator: Discriminator-free adversarial domain adaptation,” in *CVPR*, 2022, pp. 7181–7190.
- [55] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [56] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *ICLR*, 2021.
- [57] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
- [58] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer, “Scaling vision transformers,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 12 104–12 113.
- [59] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, “Learning transferable visual models from natural language supervision,” in *ICML*, 2021, pp. 8748–8763.
- [60] J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, and Y. Wu, “Coca: Contrastive captioners are image-text foundation models,” *arXiv preprint arXiv:2205.01917*, 2022.
- [61] L. Yuan, D. Chen, Y.-L. Chen, N. Codella, X. Dai, J. Gao, H. Hu, X. Huang, B. Li, C. Li, C. Liu, M. Liu, Z. Liu, Y. Lu, Y. Shi, L. Wang, J. Wang, B. Xiao, Z. Xiao, J. Yang, M. Zeng, L. Zhou, and P. Zhang, “Florence: A new foundation model for computer vision,” 2021. [Online]. Available: <https://arxiv.org/abs/2111.11432>
- [62] K. Mei, C. Zhu, J. Zou, and S. Zhang, “Instance adaptive self-training for unsupervised domain adaptation,” in *ECCV*, 2020, pp. 415–430.

- [63] Y. Zou, Z. Yu, X. Liu, B. Kumar, and J. Wang, “Confidence regularized self-training,” in *ICCV*, 2019, pp. 5982–5991.
- [64] Y. Zou, Z. Yu, B. Kumar, and J. Wang, “Unsupervised domain adaptation for semantic segmentation via class-balanced self-training,” in *ECCV*, 2018, pp. 289–305.
- [65] X. Wang, Z. Wu, L. Lian, and S. X. Yu, “Debiased learning from naturally imbalanced pseudo-labels,” in *CVPR*, 2022, pp. 14 647–14 657.
- [66] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, “Moment matching for multi-source domain adaptation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1406–1415.
- [67] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and K. Saenko, “Visda: The visual domain adaptation challenge,” *arXiv preprint arXiv:1710.06924*, 2017.
- [68] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, “Deep hashing network for unsupervised domain adaptation,” in *CVPR*, 2017, pp. 5018–5027.
- [69] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, “Adapting visual category models to new domains,” in *European conference on computer vision*. Springer, 2010, pp. 213–226.
- [70] Z. Lai, S. Vedapunt, N. Zhou, J. Wu, C. P. Huynh, X. Li, K. K. Fu, and C.-N. Chuah, “Pad-clip: Pseudo-labeling with adaptive debiasing in clip for unsupervised domain adaptation,” in *ICCV*, 2023.
- [71] W. Zhang, L. Zhu, J. Hallinan, S. Zhang, A. Makmur, Q. Cai, and B. C. Ooi, “Boostmis: Boosting medical image semi-supervised learning with adaptive pseudo labeling and informative active annotation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 20 666–20 676.
- [72] Z. Lai, C. Wang, L. C. Oliveira, B. N. Dugger, S.-C. Cheung, and C.-N. Chuah, “Joint semi-supervised and active learning for segmentation of gigapixel pathology images with cost-effective labeling,” in *ICCV Workshop*, 2021, pp. 591–600.
- [73] P. Gao, S. Geng, R. Zhang, T. Ma, R. Fang, Y. Zhang, H. Li, and Y. Qiao, “Clip-adapter: Better vision-language models with feature adapters,” *arXiv preprint arXiv:2110.04544*, 2021.
- [74] M. Liu, L. Hu, Y. Tang, C. Wang, Y. He, C. Zeng, K. Lin, Z. He, and W. Huo, “A deep learning method for breast cancer classification in the pathology images,” *IEEE J. Biomed. Health. Inf.*, vol. 26, no. 10, pp. 5025–5032, 2022.
- [75] J. Yang, H. Chen, Y. Liang, J. Huang, L. He, and J. Yao, “Concl: Concept contrastive learning for dense prediction pre-training in pathology images,” in *ECCV*. Springer, 2022, pp. 523–539.

- [76] X. Wang, S. Yang, J. Zhang, M. Wang, J. Zhang, J. Huang, W. Yang, and X. Han, “Transpath: Transformer-based self-supervised learning for histopathological image classification,” in *Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Springer, 2021, pp. 186–195.
- [77] C. Ge, R. Huang, M. Xie, Z. Lai, S. Song, S. Li, and G. Huang, “Domain adaptation via prompt learning,” *arXiv preprint arXiv:2202.06687*, 2022.
- [78] Z. Lai, Z. Li, L. Cerny Oliveira, J. Chauhan, B. Dugger, and C.-N. Chuah, “Clipath: Fine-tune clip with visual feature fusion for pathology image analysis towards minimizing data collection efforts,” in *ICCV Workshop on Computer Vision for Automated Medical Diagnosis*, 2023.
- [79] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.
- [80] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *MICCAI*. Springer, 2015, pp. 234–241.
- [81] F. Milletari, S.-A. Ahmadi, C. Kroll, A. Plate, V. Rozanski, J. Maiostre, J. Levin, O. Dietrich, B. Ertl-Wagner, K. Bötzel *et al.*, “Hough-CNN: deep learning for segmentation of deep brain regions in MRI and ultrasound,” *Computer Vision and Image Understanding*, vol. 164, pp. 92–102, 2017.
- [82] P. Coupé, J. V. Manjón, V. Fonov, J. Pruessner, M. Robles, and D. L. Collins, “Patch-based segmentation using expert priors: Application to hippocampus and ventricle segmentation,” *NeuroImage*, vol. 54, no. 2, pp. 940–954, 2011.
- [83] S. M. McKinney, M. Sieniek, V. Godbole, J. Godwin, N. Antropova, H. Ashrafiyan, T. Back, M. Chesus, G. C. Corrado, A. Darzi *et al.*, “International evaluation of an ai system for breast cancer screening,” *Nature*, vol. 577, no. 7788, pp. 89–94, 2020.
- [84] H. Hiary, R. S. Alomari, and V. Chaudhary, “Segmentation and localisation of whole slide images using unsupervised learning,” *IET Image Processing*, vol. 7, no. 5, pp. 464–471, 2013.
- [85] P. Kłeczek, G. Dyduch, J. Jaworek-Korjakowska, and R. Tadeusiewicz, “Automated epidermis segmentation in histopathological images of human skin stained with hematoxylin and eosin,” in *Medical Imaging 2017: Digital Pathology*, vol. 10140. International Society for Optics and Photonics, 2017, p. 101400M.
- [86] A. Cruz-Roa, A. Basavanahally, F. González, H. Gilmore, M. Feldman, S. Ganesan, N. Shih, J. Tomaszewski, and A. Madabhushi, “Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks,” in *Medical Imaging 2014: Digital Pathology*, vol. 9041. International Society for Optics and Photonics, 2014, p. 904103.

- [87] Z. Guo, H. Liu, H. Ni, X. Wang, M. Su, W. Guo, K. Wang, T. Jiang, and Y. Qian, "A fast and refined cancer regions segmentation framework in whole-slide breast pathological images," *Scientific reports*, vol. 9, no. 1, pp. 1–10, 2019.
- [88] N. Wahab, A. Khan, and Y. S. Lee, "Transfer learning based deep cnn for segmentation and detection of mitoses in breast cancer histopathological images," *Microscopy*, vol. 68, no. 3, pp. 216–233, 2019.
- [89] R. Wetteland, V. Kvikstad, T. Eftestøl, E. Tøssebro, M. Lillesand, E. A. Janssen, and K. Engan, "Automatic diagnostic tool for predicting cancer grade in bladder cancer patients using deep learning," *IEEE Access*, vol. 9, pp. 115 813–115 825, 2021.
- [90] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer, 2018, pp. 3–11.
- [91] X. Li, Y. Wang, Q. Tang, Z. Fan, and J. Yu, "Dual u-net for the segmentation of overlapping glioma nuclei," *IEEE Access*, vol. 7, pp. 84 040–84 052, 2019.
- [92] S. Mehta, E. Mercan, J. Bartlett, D. Weaver, J. G. Elmore, and L. Shapiro, "Y-net: joint segmentation and classification for diagnosis of breast biopsy images," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2018, pp. 893–901.
- [93] J. Li, K. V. Sarma, K. C. Ho, A. Gertych, B. S. Knudsen, and C. W. Arnold, "A multi-scale u-net for semantic segmentation of histological images from radical prostatectomies," in *AMIA Annual Symposium Proceedings*, vol. 2017. American Medical Informatics Association, 2017, p. 1140.
- [94] X. Feng, N. J. Tustison, S. H. Patel, and C. H. Meyer, "Brain tumor segmentation using an ensemble of 3d u-nets and overall survival prediction using radiomic features," *Frontiers in Computational Neuroscience*, vol. 14, p. 25, 2020. [Online]. Available: <https://www.frontiersin.org/article/10.3389/fncom.2020.00025>
- [95] M. Livne, J. Rieger, O. U. Aydin, A. A. Taha, E. M. Akay, T. Kossen, J. Sobesky, J. D. Kelleher, K. Hildebrand, D. Frey, and V. I. Madai, "A u-net deep learning framework for high performance vessel segmentation in patients with cerebrovascular disease," *Frontiers in Neuroscience*, vol. 13, p. 97, 2019. [Online]. Available: <https://www.frontiersin.org/article/10.3389/fnins.2019.00097>
- [96] B. Lee, N. Yamanakkanavar, and J. Y. Choi, "Automatic segmentation of brain mri using a novel patch-wise u-net deep architecture," *PLOS ONE*, vol. 15, no. 8, pp. 1–20, 08 2020. [Online]. Available: <https://doi.org/10.1371/journal.pone.0236493>
- [97] A. R. Vega, R. Chkheidze, V. Jarmale, P. Shang, C. Foong, M. I. Diamond, C. L. White, and S. Rajaram, "Deep learning reveals disease-specific signatures of white matter pathology in tauopathies," *Acta neuropathologica communications*, vol. 9, no. 1, pp. 1–13, 2021.

- [98] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [99] M. Hesamian, W. Jia, X. He, and P. Kennedy, "Deep learning techniques for medical image segmentation: Achievements and challenges," *J Digit Imaging* 32, pp. 582–596, 2018.
- [100] E. Gibson, F. Giganti, Y. Hu, E. Bonmati, S. Bandula, K. Gurusamy, B. R. Davidson, S. P. Pereira, M. J. Clarkson, and D. C. Barratt, "Towards image-guided pancreas and biliary endoscopy: automatic multi-organ segmentation on abdominal ct with dense dilated networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2017, p. 728–736.
- [101] E. Gibson, F. Giganti, Y. Hu, E. Bonmati, S. Bandula, K. Gurusamy, B. R. Davidson, S. P. Pereira, M. J. Clarkson, and D. C. Barratt, "Towards image-guided pancreas and biliary endoscopy: Automatic multi-organ segmentation on abdominal ct with dense dilated networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2017, pp. 728–736.
- [102] P. Hu, F. Wu, J. Peng, Y. Bao, F. Chen, and D. Kong, "Automatic abdominal multi-organ segmentation using deep convolutional neural network and time-implicit level sets," in *Radiol Surg* 12, 2017, p. 399–411.
- [103] P. F. Christ, F. Ettlinger, F. Gr, and M. E. etal, "Automatic liver and tumor segmentation of ct and mri volumes using cascaded fully convolutional neural networks," *arXiv preprint arXiv:1702.05970*, 2017.
- [104] G. Z. Guodong Zeng, "Multi-stream 3d fcn with multi-scale deep supervision for multi-modality isointense infant brain mr image segmentation," in *International Symposium on Biomedical Imaging*, 2018, p. 136–140.
- [105] X.-Y. Zhou, C. Riga, S.-L. Lee, and G.-Z. Yang, "Towards automatic 3d shape instantiation for deployed stent grafts: 2d multiple-class and class-imbalance marker segmentation with equally-weighted focal u-net," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.
- [106] S. L. XiaoYun Zhou, Mali Shen, "Focal fcn: towards small object segmentation with limited training data," *arXiv preprint arXiv:1711.01506*, 2017.
- [107] H. Dong, G. Yang, F. Liu, Y. Mo, and Y. Guo, "Automatic brain tumor detection and segmentation using u-net based fully convolutional networks," in *Annual Conference on Medical Image Understanding and Analysis*. Springer, 2017, pp. 506–517.
- [108] A. Rampun, D. Jarvis, P. Griffiths, and P. Armitage, "Automated 2d fetal brain segmentation of mr images using a deep u-net," in *Asian Conference on Pattern Recognition*. Springer, 2019, pp. 373–386.

- [109] R. K. Pandey, A. Vasan, and A. Ramakrishnan, "Segmentation of liver lesions with reduced complexity deep models," *arXiv preprint arXiv:1805.09233*, 2018.
- [110] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.
- [111] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: <http://arxiv.org/abs/1502.03167>
- [112] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, p. 1929–1958, Jan. 2014.
- [113] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *CoRR*, vol. abs/1502.01852, 2015. [Online]. Available: <http://arxiv.org/abs/1502.01852>
- [114] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [115] D. A. Gutman, M. Khalilia, S. Lee, M. Nalisnik, Z. Mullen, J. Beezley, D. R. Chittajallu, D. Manthey, and L. A. Cooper, "The digital slide archive: A software platform for management, integration, and analysis of histology for cancer research," *Cancer research*, vol. 77, no. 21, pp. e75–e78, 2017.
- [116] D. Wang, A. Khosla, R. Gargeya, H. Irshad, and A. H. Beck, "Deep learning for identifying metastatic breast cancer," *arXiv preprint arXiv:1606.05718*, 2016.
- [117] L. Hou, D. Samaras, T. M. Kurc, Y. Gao, J. E. Davis, and J. H. Saltz, "Patch-based convolutional neural network for whole slide tissue image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2424–2433.
- [118] Y. S. Vang, Z. Chen, and X. Xie, "Deep learning framework for multi-class breast cancer histology image classification," in *International Conference Image Analysis and Recognition*. Springer, 2018, pp. 914–922.
- [119] F. G. Zanjani, S. Zinger *et al.*, "Cancer detection in histopathology whole-slide images using conditional random fields on deep embedded spaces," in *Medical imaging 2018: Digital Pathology*, vol. 10581. International Society for Optics and Photonics, 2018, p. 105810I.
- [120] B. Kong, X. Wang, Z. Li, Q. Song, and S. Zhang, "Cancer metastasis detection via spatially structured deep network," in *International Conference on Information Processing in Medical Imaging*. Springer, 2017, pp. 236–248.

- [121] T. Artieres *et al.*, “Neural conditional random fields,” in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 177–184.
- [122] J. F. Poduslo, G. L. Curran, J. A. Peterson, D. J. McCormick, A. H. Fauq, M. A. Khan, and T. M. Wengenack, “Design and chemical synthesis of a magnetic resonance contrast agent with enhanced in vitro binding, high blood–brain barrier permeability, and in vivo targeting to alzheimer’s disease amyloid plaques,” *Biochemistry*, vol. 43(20), pp. 6064–6075, 05 2004.
- [123] K. Martinez and J. Cupitt, “Libvips: A fast image processing library with low memory needs,” 2007.
- [124] A. Khan, A. Sohail, U. Zahoor, and A. S. Qureshi, “A survey of the recent architectures of deep convolutional neural networks,” *arXiv preprint arXiv:1901.06032*, 2019.
- [125] D. McNeely-White, J. R. Beveridge, and B. A. Draper, “Inception and resnet features are (almost) equivalent,” *Cognitive Systems Research*, vol. 59, pp. 312–318, 2020.
- [126] P. Krähenbühl and V. Koltun, “Efficient inference in fully connected crfs with gaussian edge potentials,” in *Advances in neural information processing systems*, 2011, pp. 109–117.
- [127] Y. Li and W. Ping, “Cancer metastasis detection with neural conditional random field,” in *Medical Imaging with Deep Learning*, 2018.
- [128] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr, “Conditional random fields as recurrent neural networks,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1529–1537.
- [129] K. H. Zou, S. K. Warfield, A. Bharatha, C. M. Tempany, M. R. Kaus, S. J. Haker, W. M. Wells III, F. A. Jolesz, and R. Kikinis, “Statistical validation of image segmentation quality based on a spatial overlap index1: scientific reports,” *Academic radiology*, vol. 11, no. 2, pp. 178–189, 2004.
- [130] T. Rahman, A. Khandakar, M. A. Kadir, K. R. Islam, K. F. Islam, R. Mazhar, T. Hamid, M. T. Islam, S. Kashem, Z. B. Mahbub *et al.*, “Reliable tuberculosis detection using chest x-ray with deep learning, segmentation and visualization,” *IEEE Access*, vol. 8, pp. 191 586–191 601, 2020.
- [131] I. Ahmed, M. Ahmad, F. A. Khan, and M. Asif, “Comparison of deep-learning-based segmentation models: Using top view person images,” *IEEE Access*, vol. 8, pp. 136 361–136 373, 2020.
- [132] J. Lu and K.-y. Tong, “Visualized insights into the optimization landscape of fully convolutional networks,” *arXiv preprint arXiv:1901.08556*, 2019.
- [133] K. V. Chuang and M. J. Keiser, “Adversarial controls for scientific machine learning,” 2018.

- [134] B. N. Dugger and D. W. Dickson, "Pathology of neurodegenerative diseases," *Cold Spring Harbor perspectives in biology*, vol. 9, no. 7, p. a028035, 2017.
- [135] A. Serrano-Pozo, M. P. Frosch, E. Masliah, and B. T. Hyman, "Neuropathological alterations in alzheimer disease," *Cold Spring Harbor perspectives in medicine*, vol. 1, no. 1, p. a006189, 2011.
- [136] D. W. Dickson, "The pathogenesis of senile plaques," *Journal of Neuropathology & Experimental Neurology*, vol. 56, no. 4, pp. 321–339, 1997.
- [137] N. Dimitriou, O. Arandjelović, and P. D. Caie, "Deep learning for whole slide image analysis: an overview," *Frontiers in medicine*, vol. 6, p. 264, 2019.
- [138] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations (ICLR)*, 2015.
- [139] L. Chan, M. S. Hosseini, C. Rowsell, K. N. Plataniotis, and S. Damaskinos, "Histosegnet: Semantic segmentation of histological tissue type in whole slide images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, 2019, pp. 10 662–10 671.
- [140] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- [141] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness, "Pseudo-labeling and confirmation bias in deep semi-supervised learning," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.
- [142] S. Calderon-Ramirez, A. Moemeni, D. Elizondo, S. Colreavy-Donnelly, L. F. Chavarria-Estrada, M. A. Molina-Cabello *et al.*, "Correcting data imbalance for semi-supervised covid-19 detection using x-ray chest images," *arXiv preprint arXiv:2008.08496*, 2020.
- [143] A. Krizhevsky, "Learning multiple layers of features from tiny images," Department of Computer Science, University of Toronto, Tech. Rep., 2009.
- [144] Q. Dong, S. Gong, and X. Zhu, "Imbalanced deep learning by minority class incremental rectification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.
- [145] Y. Yang and Z. Xu, "Rethinking the value of labels for improving class-imbalanced learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 19 290–19 301.
- [146] M. Hyun, J. Jeong, and N. Kwak, "Class-imbalanced semi-supervised learning," *arXiv preprint arXiv:2002.06815*, 2020.
- [147] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning (ICML)*, 2013.



- [148] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, “Unsupervised data augmentation for consistency training,” *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020.
- [149] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, “Self-training with noisy student improves imagenet classification,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10 687–10 698.
- [150] M. Sajjadi, M. Javanmardi, and T. Tasdizen, “Regularization with stochastic transformations and perturbations for deep semi-supervised learning,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.
- [151] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, “Virtual adversarial training: a regularization method for supervised and semi-supervised learning,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1979–1993, 2018.
- [152] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu, “Large-scale long-tailed recognition in an open world,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2537–2546.
- [153] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie, “The inaturalist species classification and detection dataset,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [154] M. A. Jamal, M. Brown, M.-H. Yang, L. Wang, and B. Gong, “Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 7610–7619.
- [155] J. Liu, Y. Sun, C. Han, Z. Dou, and W. Li, “Deep representation learning on long-tailed data: A learnable embedding augmentation perspective,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2970–2979.
- [156] Y.-X. Wang, D. Ramanan, and M. Hebert, “Learning to model the tail,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- [157] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: synthetic minority over-sampling technique,” *Journal of Artificial Intelligence Research*, 2002.
- [158] M. Buda, A. Maki, and M. A. Mazurowski, “A systematic study of the class imbalance problem in convolutional neural networks,” *Neural Networks*, vol. 106, pp. 249–259, 2018.
- [159] S. Pouyanfar, Y. Tao, A. Mohan, H. Tian, A. S. Kaseb, K. Gauen, R. Dailey, S. Aghajanzadeh, Y.-H. Lu, S.-C. Chen *et al.*, “Dynamic sampling in convolutional neural networks for imbalanced data classification,” in *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*. IEEE, 2018, pp. 112–117.

- [160] Y. S. Aurelio, G. M. de Almeida, C. L. de Castro, and A. P. Braga, “Learning from imbalanced data sets with weighted cross-entropy function,” *Neural Processing Letters*, vol. 50, no. 2, pp. 1937–1949, 2019.
- [161] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, “Learning imbalanced datasets with label-distribution-aware margin loss,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.
- [162] J. Tan, C. Wang, B. Li, Q. Li, W. Ouyang, C. Yin, and J. Yan, “Equalization loss for long-tailed object recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 662–11 671.
- [163] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, “Cost-sensitive learning of deep feature representations from imbalanced data,” *IEEE transactions on neural networks and learning systems*, vol. 29, no. 8, pp. 3573–3587, 2017.
- [164] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, “Class-balanced loss based on effective number of samples,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [165] J. Kim, J. Jeong, and J. Shin, “M2m: Imbalanced classification via major-to-minor translation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 13 896–13 905.
- [166] J. Byrd and Z. Lipton, “What is the effect of importance weighting in deep learning?” in *International Conference on Machine Learning*. PMLR, 2019, pp. 872–881.
- [167] C. Huang, Y. Li, C. Change Loy, and X. Tang, “Learning deep representation for imbalanced classification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [168] P. Branco, L. Torgo, and R. P. Ribeiro, “A survey of predictive modeling on imbalanced domains,” *ACM Computing Surveys (CSUR)*, vol. 49, no. 2, p. 31, 2016.
- [169] M. Kubat, “Addressing the curse of imbalanced training sets: One-sided selection,” in *Proceedings of the International Conference on Machine Learning (ICML)*, 1997, pp. 179–186.
- [170] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, “Learning imbalanced datasets with label-distribution-aware margin loss,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
- [171] A. K. Menon, S. Jayasumana, A. S. Rawat, H. Jain, A. Veit, and S. Kumar, “Long-tail learning via logit adjustment,” in *International Conference on Learning Representations*, 2020.
- [172] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis, “Decoupling representation and classifier for long-tailed recognition,” in *International Conference on Learning Representations (ICLR)*, 2020.

- [173] B. Zhou, Q. Cui, X.-S. Wei, and Z.-M. Chen, “Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9719–9728.
- [174] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis, “Decoupling representation and classifier for long-tailed recognition,” in *International Conference on Learning Representations*, 2019.
- [175] Z.-H. Zhou and X.-Y. Liu, “Training cost-sensitive neural networks with methods addressing the class imbalance problem,” *IEEE Transactions on knowledge and data engineering*, vol. 18, no. 1, pp. 63–77, 2005.
- [176] A. Coates, A. Ng, and H. Lee, “An analysis of single-layer networks in unsupervised feature learning,” in *Proceedings of the fourteenth international conference on artificial intelligence and statistics (AISTATS)*, 2011.
- [177] N. Japkowicz, “The class imbalance problem: Significance and strategies,” in *Proceedings of the International Conference on Artificial Intelligence (ICAI)*, 2000.
- [178] R. Busa-Fekete, B. Szörényi, K. Dembczynski, and E. Hüllermeier, “Online F-measure optimization,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 28, 2015.
- [179] N. Houlsby, J. M. Hernández-Lobato, and Z. Ghahramani, “Cold-start active learning with robust ordinal matrix factorization,” in *International Conference on Machine Learning*. PMLR, 2014, pp. 766–774.
- [180] S. C. Hoi, R. Jin, J. Zhu, and M. R. Lyu, “Batch mode active learning and its application to medical image classification,” in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 417–424.
- [181] O. Sener and S. Savarese, “Active learning for convolutional neural networks: A core-set approach,” in *International Conference on Learning Representations*, 2018.
- [182] J. T. Ash, C. Zhang, A. Krishnamurthy, J. Langford, and A. Agarwal, “Deep batch active learning by diverse, uncertain gradient lower bounds,” in *International Conference on Learning Representations*, 2019.
- [183] A. Kirsch, J. Van Amersfoort, and Y. Gal, “Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning,” *Advances in neural information processing systems*, vol. 32, pp. 7026–7037, 2019.
- [184] K. Wei, R. Iyer, and J. Bilmes, “Submodularity in data subset selection and active learning,” in *International Conference on Machine Learning*. PMLR, 2015, pp. 1954–1963.
- [185] K. Killamsetty, D. Sivasubramanian, G. Ramakrishnan, and R. Iyer, “Glister: Generalization based data subset selection for efficient and robust learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 9, 2021, pp. 8110–8118.

- [186] J. E. Van Engelen and H. H. Hoos, “A survey on semi-supervised learning,” *Machine Learning*, vol. 109, no. 2, pp. 373–440, 2020.
- [187] C.-W. Kuo, C.-Y. Ma, J.-B. Huang, and Z. Kira, “Featmatch: Feature-based augmentation for semi-supervised learning,” in *European Conference on Computer Vision*. Springer, 2020, pp. 479–495.
- [188] S. Budd, E. C. Robinson, and B. Kainz, “A survey on active learning and human-in-the-loop deep learning for medical image analysis,” *Medical Image Analysis*, p. 102062, 2021.
- [189] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, X. Chen, and X. Wang, “A survey of deep active learning,” *arXiv preprint arXiv:2009.00236*, 2020.
- [190] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, “Multi-class active learning for image classification,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 2372–2379.
- [191] S. Fine, R. Gilad-Bachrach, and E. Shamir, “Query by committee, linear separation and random walks,” *Theoretical Computer Science*, vol. 284, no. 1, pp. 25–51, 2002.
- [192] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby, “Selective sampling using the query by committee algorithm,” *Machine learning*, vol. 28, no. 2, pp. 133–168, 1997.
- [193] T. Graepel and R. Herbrich, “The kernel gibbs sampler,” *Advances in Neural Information Processing Systems*, vol. 13, pp. 514–520, 2000.
- [194] H. S. Seung, M. Opper, and H. Sompolinsky, “Query by committee,” in *Proceedings of the fifth annual workshop on Computational learning theory*, 1992, pp. 287–294.
- [195] I. Campbell, N. Cristianini, and A. Smola, “Query learning with large margin classifiers,” in *Proc. 17th Int Conference on Machine Learning*. Morgan Kaufmann, 2000, pp. 111–118.
- [196] G. SCHON, “Less is more: Active learning with support vector machines,” in *Proc. Seventeenth International Conference on Machine Learning, 2000*, 2000, pp. 839–846.
- [197] S. Tong and D. Koller, “Support vector machine active learning with applications to text classification,” *Journal of machine learning research*, vol. 2, no. Nov, pp. 45–66, 2001.
- [198] T. Drugman, J. Pylkkonen, and R. Kneser, “Active and semi-supervised learning in asr: Benefits on the acoustic and language models,” *arXiv preprint arXiv:1903.02852*, 2019.
- [199] P. K. Rhee, E. Erdene, S. D. Kyun, M. U. Ahmed, and S. Jin, “Active and semi-supervised learning for object detection with imperfect data,” *Cognitive Systems Research*, vol. 45, pp. 109–123, 2017.

- [200] M. Gao, Z. Zhang, G. Yu, S. Ö. Arik, L. S. Davis, and T. Pfister, “Consistency-based semi-supervised active learning: Towards minimizing labeling cost,” in *European Conference on Computer Vision*. Springer, 2020, pp. 510–526.
- [201] B. Cheng, A. G. Schwing, and A. Kirillov, “Per-pixel classification is not all you need for semantic segmentation,” 2021.
- [202] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, “Randaugment: Practical automated data augmentation with a reduced search space,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 702–703.
- [203] T. Wang, S. Chen, and R. Jia, “One-round active learning,” *arXiv preprint arXiv:2104.11843*, 2021.
- [204] R. Feng, X. Liu, J. Chen, D. Z. Chen, H. Gao, and J. Wu, “A deep learning approach for colonoscopy pathology wsi analysis: accurate segmentation and classification,” *IEEE J. Biomed. Health. Inf.*, vol. 25, no. 10, pp. 3700–3708, 2020.
- [205] S. Graham, D. Epstein, and N. Rajpoot, “Dense steerable filter cnns for exploiting rotational symmetry in histology images,” *IEEE Trans. Med. Imaging*, vol. 39, no. 12, pp. 4124–4136, 2020.
- [206] H. Pinckaers, W. Bulten, J. van der Laak, and G. Litjens, “Detection of prostate cancer in whole-slide images through end-to-end training with image-level labels,” *IEEE Trans. Med. Imaging*, vol. 40, no. 7, pp. 1817–1826, 2021.
- [207] N. A. Koohbanani, B. Unnikrishnan, S. A. Khurram, P. Krishnaswamy, and N. Rajpoot, “Self-path: Self-supervision for classification of pathology images with limited annotations,” *IEEE Trans. Med. Imaging*, vol. 40, no. 10, pp. 2845–2856, 2021.
- [208] C. You, W. Dai, Y. Min, F. Liu, X. Zhang, C. Feng, D. A. Clifton, S. K. Zhou, L. H. Staib, and J. S. Duncan, “Rethinking semi-supervised medical image segmentation: A variance-reduction perspective,” *arXiv preprint arXiv:2302.01735*, 2023.
- [209] C. You, W. Dai, F. Liu, Y. Min, H. Su, X. Zhang, X. Li, D. A. Clifton, L. Staib, and J. S. Duncan, “Mine your own anatomy: Revisiting medical image segmentation with extremely limited labels,” *arXiv preprint arXiv:2209.13476*, 2022.
- [210] Z. Zhao, Z. Zeng, K. Xu, C. Chen, and C. Guan, “Dsal: Deeply supervised active learning from strong and weak labelers for biomedical image segmentation,” *IEEE J. Biomed. Health. Inf.*, vol. 25, no. 10, pp. 3744–3751, 2021.
- [211] D. Mahapatra, P. J. Schöffler, J. A. Tielbeek, F. M. Vos, and J. M. Buhmann, “Semi-supervised and active learning for automatic segmentation of crohn’s disease,” in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013: 16th International Conference, Nagoya, Japan, September 22–26, 2013, Proceedings, Part II 16*. Springer, 2013, pp. 214–221.

- [212] Y. Xu, L. Shang, J. Ye, Q. Qian, Y.-F. Li, B. Sun, H. Li, and R. Jin, “Dash: Semi-supervised learning with dynamic thresholding,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 11 525–11 536.
- [213] J. Choi, I. Elezi, H.-J. Lee, C. Farabet, and J. M. Alvarez, “Active learning for deep object detection via probabilistic modeling,” in *ICCV*, 2021, pp. 10 264–10 273.
- [214] K. Kim, D. Park, K. I. Kim, and S. Y. Chun, “Task-aware variational adversarial active learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8166–8175.
- [215] D. Gudovskiy, A. Hodgkinson, T. Yamaguchi, and S. Tsukizawa, “Deep active learning for biased datasets via fisher kernel self-supervision,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9041–9049.
- [216] M. Jahanifar, N. Z. Tajeddin, N. A. Koohbanani, and N. M. Rajpoot, “Robust interactive semantic segmentation of pathology images with minimal user input,” in *ICCV*, 2021, pp. 674–683.
- [217] R. H. Myers, “Elementary applied statistics,” 1974.
- [218] Y. Gal, R. Islam, and Z. Ghahramani, “Deep bayesian active learning with image data,” in *International Conference on Machine Learning*. PMLR, 2017, pp. 1183–1192.
- [219] F. Peng, C. Wang, J. Liu, and Z. Yang, “Active learning for lane detection: A knowledge distillation approach,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15 152–15 161.
- [220] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *ICLR (Poster)*, 2015.
- [221] B. S. Veeling, J. Linmans, J. Winkens, T. Cohen, and M. Welling, “Rotation equivariant cnns for digital pathology,” in *Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Springer, 2018, pp. 210–218.
- [222] J. Wei, A. Suriawinata, B. Ren, X. Liu, M. Lisovsky, L. Vaickus, C. Brown, M. Baker, N. Tomita, L. Torresani *et al.*, “A petri dish for histopathology image analysis,” in *Int. Conf. Arti. Intell. in Medi.* Springer, 2021, pp. 11–24.
- [223] H. Rangwani, S. K. Aithal, M. Mishra, A. Jain, and V. B. Radhakrishnan, “A closer look at smoothness in domain adversarial training,” in *ICML*, 2022, pp. 18 378–18 399.
- [224] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, “Deep domain confusion: Maximizing for domain invariance,” *arXiv preprint arXiv:1412.3474*, 2014.
- [225] M. Long, H. Zhu, J. Wang, and M. I. Jordan, “Deep transfer learning with joint adaptation networks,” in *ICML*, 2017, pp. 2208–2217.
- [226] B. Sun and K. Saenko, “Deep coral: Correlation alignment for deep domain adaptation,” in *ECCV*, 2016, pp. 443–450.

- [227] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann, “Contrastive adaptation network for unsupervised domain adaptation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4893–4902.
- [228] M. Long, Z. Cao, J. Wang, and M. I. Jordan, “Conditional adversarial domain adaptation,” in *NeurIPS*, 2018, pp. 1645–1655.
- [229] S. Lee, D. Kim, N. Kim, and S.-G. Jeong, “Drop to adapt: Learning discriminative features for unsupervised domain adaptation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 91–100.
- [230] S. Yang, Y. Wang, K. Wang, S. Jui *et al.*, “Attracting and dispersing: A simple approach for source-free domain adaptation,” in *Advances in Neural Information Processing Systems*, 2022.
- [231] S. Hur, I. Shin, K. Park, S. Woo, and I. S. Kweon, “Learning classifiers of prototypes and reciprocal points for universal domain adaptation,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 531–540.
- [232] J. Yang, J. Liu, N. Xu, and J. Huang, “Tvt: Transferable vision transformer for unsupervised domain adaptation,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 520–530.
- [233] Y.-J. Li, X. Dai, C.-Y. Ma, Y.-C. Liu, K. Chen, B. Wu, Z. He, K. Kitani, and P. Vajda, “Cross-domain adaptive teacher for object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7581–7590.
- [234] H. Liu, J. Wang, and M. Long, “Cycle self-training for domain adaptation,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 22 968–22 981, 2021.
- [235] J. Kim, K. Ryoo, J. Seo, G. Lee, D. Kim, H. Cho, and S. Kim, “Semi-supervised learning of semantic correspondence with pseudo-labels,” in *CVPR*, 2022, pp. 19 699–19 709.
- [236] T. Sun, C. Lu, and H. Ling, “Prior knowledge guided unsupervised domain adaptation,” in *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*. Springer, 2022, pp. 639–655.
- [237] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig, “Scaling up visual and vision-language representation learning with noisy text supervision,” in *ICML*, 2021, pp. 4904–4916.
- [238] X. Zhai, X. Wang, B. Mustafa, A. Steiner, D. Keysers, A. Kolesnikov, and L. Beyer, “Lit: Zero-shot transfer with locked-image text tuning,” in *CVPR*, 2022, pp. 18 123–18 133.
- [239] Y. Lu, J. Liu, Y. Zhang, Y. Liu, and X. Tian, “Prompt distribution learning,” in *CVPR*, 2022, pp. 5206–5215.
- [240] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *NeurIPS*, vol. 30, 2017.

- [241] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, “Learning to prompt for vision-language models,” *Int. J. Comput. Vis.*, pp. 1–12, 2022.
- [242] Z. Dong, Z. Hanwang, T. Jinhui, H. Xiansheng, and S. Qianru, “Causal intervention for weakly supervised semantic segmentation,” in *NeurIPS*, 2020.
- [243] D. Zhang, H. Zhang, J. Tang, X.-S. Hua, and Q. Sun, “Causal intervention for weakly-supervised semantic segmentation,” *NeurIPS*, vol. 33, pp. 655–666, 2020.
- [244] J. Qi, Y. Niu, J. Huang, and H. Zhang, “Two causal principles for improving visual dialog,” in *CVPR*, 2020, pp. 10 860–10 869.
- [245] K. Tang, Y. Niu, J. Huang, J. Shi, and H. Zhang, “Unbiased scene graph generation from biased training,” in *CVPR*, 2020, pp. 3716–3725.
- [246] B. Liu, D. Wang, X. Yang, Y. Zhou, R. Yao, Z. Shao, and J. Zhao, “Show, deconfound and tell: Image captioning with causal inference,” in *CVPR*, 2022, pp. 18 041–18 050.
- [247] J. Huang, Y. Qin, J. Qi, Q. Sun, and H. Zhang, “Deconfounded visual grounding,” in *AAAI*, 2022, pp. 998–1006.
- [248] Z. Chen, Z. Tian, J. Zhu, C. Li, and S. Du, “C-cam: Causal cam for weakly supervised semantic segmentation on medical image,” in *CVPR*, 2022, pp. 11 676–11 685.
- [249] A. Agrawal, J. Lu, S. Antol, M. Mitchell, C. L. Zitnick, D. Batra, and D. Parikh, “Vqa: Visual question answering,” 2015. [Online]. Available: <https://arxiv.org/abs/1505.00468>
- [250] D. Lopez-Paz, R. Nishihara, S. Chintala, B. Scholkopf, and L. Bottou, “Discovering causal signals in images,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [251] Z. Yue, H. Zhang, Q. Sun, and X.-S. Hua, “Interventional few-shot learning,” in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 2734–2746. [Online]. Available: <https://proceedings.neurips.cc/paper/2020/file/1cc8a8ea51cd0adddf5dab504a285915-Paper.pdf>
- [252] T. Wang, J. Huang, H. Zhang, and Q. Sun, “Visual commonsense r-cnn,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [253] X. Yang, H. Zhang, and J. Cai, “Deconfounded image captioning: A causal retrospect,” 2020. [Online]. Available: <https://arxiv.org/abs/2003.03923>
- [254] Y. Niu, K. Tang, H. Zhang, Z. Lu, X.-S. Hua, and J.-R. Wen, “Counterfactual vqa: A cause-effect look at language bias,” in *CVPR*, 2021, pp. 12 700–12 710.
- [255] M. Besserve, A. Mehrjou, R. Sun, and B. Schölkopf, “Counterfactuals uncover the modular structure of deep generative models,” in *ICLR*, 2019.



- [256] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey, “Symmetric cross entropy for robust learning with noisy labels,” in *IEEE International Conference on Computer Vision*, 2019.
- [257] X. Wang and G.-J. Qi, “Contrastive learning with stronger augmentations,” *arXiv preprint arXiv:2104.07713*, 2021.
- [258] K. Tang, J. Huang, and H. Zhang, “Long-tailed classification by keeping the good and removing the bad momentum causal effect,” *NeurIPS*, vol. 33, pp. 1513–1524, 2020.
- [259] L. Richiardi, R. Bellocco, and D. Zugna, “Mediation analysis in epidemiology: methods, interpretation and bias,” *International journal of epidemiology*, vol. 42, no. 5, pp. 1511–1519, 2013.
- [260] M. Glymour, J. Pearl, and N. P. Jewell, *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.
- [261] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh *et al.*, “Mixed precision training,” in *ICLR*, 2018.
- [262] I. Loshchilov and F. Hutter, “Sgdr: Stochastic gradient descent with warm restarts,” *arXiv preprint arXiv:1608.03983*, 2016.
- [263] G. Ilharco, M. Wortsman, R. Wightman, C. Gordon, N. Carlini, R. Taori, A. Dave, V. Shankar, H. Namkoong, J. Miller, H. Hajishirzi, A. Farhadi, and L. Schmidt, “Openclip,” Jul. 2021, if you use this software, please cite it as below. [Online]. Available: <https://doi.org/10.5281/zenodo.5143773>
- [264] P. Sharma, N. Ding, S. Goodman, and R. Soricut, “Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 2556–2565. [Online]. Available: <https://aclanthology.org/P18-1238>
- [265] Z. Du, J. Li, H. Su, L. Zhu, and K. Lu, “Cross-domain gradient discrepancy minimization for unsupervised domain adaptation,” in *CVPR*, 2021, pp. 3937–3946.
- [266] H. Xia, T. Jing, and Z. Ding, “Maximum structural generation discrepancy for unsupervised domain adaptation,” *PAMI*, 2022.
- [267] J. Huang, D. Guan, A. Xiao, S. Lu, and L. Shao, “Category contrast for unsupervised domain adaptation in visual tasks,” in *CVPR*, 2022, pp. 1203–1214.
- [268] J. Zhang, J. Huang, Z. Tian, and S. Lu, “Spectral unsupervised domain adaptation for visual recognition,” in *CVPR*, 2022, pp. 9829–9840.
- [269] R. Xu, G. Li, J. Yang, and L. Lin, “Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation,” in *ICCV*, 2019, pp. 1426–1435.

- [270] C.-Y. Lee, T. Batra, M. H. Baig, and D. Ulbricht, “Sliced wasserstein discrepancy for unsupervised domain adaptation,” in *CVPR*, 2019, pp. 10 285–10 295.
- [271] J. Zhu, H. Bai, and L. Wang, “Patch-mix transformer for unsupervised domain adaptation: A game perspective,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 3561–3571.
- [272] X. Wang, Y. Jin, M. Long, J. Wang, and M. Jordan, “Transferable normalization: Towards improving transferability of deep neural networks,” in *NeurIPS*, 2019.
- [273] J. Gao, Y. Hua, G. Hu, C. Wang, and N. M. Robertson, “Reducing distributional uncertainty by mutual information maximisation and transferable feature learning,” in *ECCV*. Springer, 2020, pp. 587–605.
- [274] R. Zhang, W. Zhang, R. Fang, P. Gao, K. Li, J. Dai, Y. Qiao, and H. Li, “Tip-adapter: Training-free adaption of clip for few-shot classification,” in *ECCV*. Springer, 2022, pp. 493–510.
- [275] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby, “Big transfer (bit): General visual representation learning,” 2019. [Online]. Available: <https://arxiv.org/abs/1912.11370>
- [276] P. Xu, X. Zhu, and D. A. Clifton, “Multimodal learning with transformers: A survey,” *arXiv preprint arXiv:2206.06488*, 2022.
- [277] Y. Tian, S. Newsam, and K. Boakye, “Fashion image retrieval with text feedback by additive attention compositional learning,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 1011–1021.
- [278] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” in *ICLR*, 2021.
- [279] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [280] H. Rasheed, M. U. khattak, M. Maaz, S. Khan, and F. S. Khan, “Finetuned clip models are efficient video learners,” in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.
- [281] R. Zhang, Z. Guo, W. Zhang, K. Li, X. Miao, B. Cui, Y. Qiao, P. Gao, and H. Li, “Pointclip: Point cloud understanding by clip,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8552–8562.
- [282] R. Gal, O. Patashnik, H. Maron, A. H. Bermanto, G. Chechik, and D. Cohen-Or, “Styleganada: Clip-guided domain adaptation of image generators,” *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, pp. 1–13, 2022.

- [283] Z. Wang, Z. Wu, D. Agarwal, and J. Sun, “Medclip: Contrastive learning from unpaired medical images and text,” *Proceedings of EMNLP*, 2022.
- [284] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *arXiv preprint arXiv:1807.03748*, 2018.
- [285] N. Houlsby, A. Giurciu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, “Parameter-efficient transfer learning for nlp,” in *ICML*. PMLR, 2019, pp. 2790–2799.
- [286] G. He, J. Chen, and J. Zhu, “Preserving pre-trained features helps calibrate fine-tuned language models,” in *ICLR*, 2023.
- [287] H. Yuan, Z. Yuan, C. Tan, F. Huang, and S. Huang, “Hype: Better pre-trained language model fine-tuning with hidden representation perturbation,” *arXiv preprint arXiv:2212.08853*, 2022.
- [288] T. Zhang, F. Wu, A. Katiyar, K. Q. Weinberger, and Y. Artzi, “Revisiting few-sample bert fine-tuning,” in *ICLR*, 2021.
- [289] C. Wu, F. Wu, T. Qi, and Y. Huang, “Noisy tune: A little noise can help you finetune pretrained language models better,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2022, pp. 680–685.
- [290] F. Chen, H. Zhang, Z. Li, J. Dou, S. Mo, H. Chen, Y. Zhang, U. Ahmed, C. Zhu, and M. Savvides, “Unitail: Detecting, reading, and matching in retail scene,” in *ECCV*. Springer, 2022, pp. 705–722.
- [291] B. E. Bejnordi, M. Veta, P. J. Van Diest, B. Van Ginneken, N. Karssemeijer, G. Litjens, J. A. Van Der Laak, M. Hermesen, Q. F. Manson, M. Balkenhol *et al.*, “Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer,” *Jama*, vol. 318, no. 22, pp. 2199–2210, 2017.
- [292] D. Liu, Y. Cui, L. Yan, C. Mousas, B. Yang, and Y. Chen, “Densernet: Weakly supervised visual localization using multi-scale feature aggregation,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 7, 2021, pp. 6101–6109.
- [293] L. Van der Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of machine learning research*, vol. 9, no. 11, 2008.
- [294] M. Khened, A. Kori, H. Rajkumar, G. Krishnamurthi, and B. Srinivasan, “A generalized deep learning framework for whole-slide image segmentation and analysis,” *Scientific reports*, vol. 11, no. 1, pp. 1–14, 2021.
- [295] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, “Meta-transfer learning for few-shot learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 403–412.

- [296] X. Li, H. Xiong, H. An, C.-Z. Xu, and D. Dou, “Rifle: Backpropagation in depth for deep transfer learning through re-initializing the fully-connected layer,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 6010–6019.
- [297] H. B. Ammar, E. Eaton, M. E. Taylor, D. C. Mocanu, K. Driessens, G. Weiss, and K. Tuyls, “An automated measure of mdp similarity for transfer in reinforcement learning,” in *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.
- [298] E. Eaton, T. Lane *et al.*, “Modeling transfer relationships between learning tasks for improved inductive transfer,” in *Joint european conference on machine learning and knowledge discovery in databases*. Springer, 2008, pp. 317–332.
- [299] J. Sinapov, S. Narvekar, M. Leonetti, and P. Stone, “Learning inter-task transferability in the absence of target task samples,” in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, 2015, pp. 725–733.
- [300] C. Nguyen, T. Hassner, M. Seeger, and C. Archambeau, “Leap: A new measure to evaluate transferability of learned representations,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 7294–7305.
- [301] Y. Tan, Y. Li, and S.-L. Huang, “Otce: A transferability metric for cross-domain cross-task representations,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 779–15 788.
- [302] S. Ben-David and R. Schuller, “Exploiting task relatedness for multiple task learning,” in *Learning theory and kernel machines*. Springer, 2003, pp. 567–580.
- [303] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, “A theory of learning from different domains,” *Machine learning*, vol. 79, no. 1, pp. 151–175, 2010.
- [304] Y. Mansour, M. Mohri, and A. Rostamizadeh, “Domain adaptation: Learning bounds and algorithms,” *arXiv preprint arXiv:0902.3430*, 2009.
- [305] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” *Advances in Neural Information Processing Systems*, vol. 27, pp. 3320–3328, 2014.
- [306] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese, “Taskonomy: Disentangling task transfer learning,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3712–3722.
- [307] Y. Bao, Y. Li, S.-L. Huang, L. Zhang, L. Zheng, A. Zamir, and L. Guibas, “An information-theoretic approach to transferability in task transfer learning,” in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 2309–2313.
- [308] A. T. Tran, C. V. Nguyen, and T. Hassner, “Transferability and hardness of supervised classification tasks,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1395–1405.

- [309] D. Jakubovitz, M. R. Rodrigues, and R. Giryes, “Lautum regularization for semi-supervised transfer learning,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.
- [310] A. Abuduweili, X. Li, H. Shi, C.-Z. Xu, and D. Dou, “Adaptive consistency regularization for semi-supervised transfer learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6923–6932.
- [311] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, “A continual learning survey: Defying forgetting in classification tasks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [312] J. Schwarz, W. Czarnecki, J. Luketina, A. Grabska-Barwinska, Y. W. Teh, R. Pascanu, and R. Hadsell, “Progress & compress: A scalable framework for continual learning,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 4528–4537.
- [313] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu, “Large scale incremental learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 374–382.
- [314] L. Yu, B. Twardowski, X. Liu, L. Herranz, K. Wang, Y. Cheng, S. Jui, and J. v. d. Weijer, “Semantic drift compensation for class-incremental learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6982–6991.
- [315] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, “Overcoming catastrophic forgetting in neural networks,” *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.
- [316] Z. Li and D. Hoiem, “Learning without forgetting,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.
- [317] F. Zenke, B. Poole, and S. Ganguli, “Continual learning through synaptic intelligence,” in *International Conference on Machine Learning*. PMLR, 2017, pp. 3987–3995.
- [318] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara, “Dark experience for general continual learning: a strong, simple baseline,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- [319] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, “icarl: Incremental classifier and representation learning,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 2001–2010.
- [320] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, “Progressive neural networks,” *arXiv preprint arXiv:1606.04671*, 2016.

- [321] X. Li, Y. Zhou, T. Wu, R. Socher, and C. Xiong, “Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 3925–3934.
- [322] M. McCloskey and N. J. Cohen, “Catastrophic interference in connectionist networks: The sequential learning problem,” in *Psychology of learning and motivation*. Elsevier, 1989, vol. 24, pp. 109–165.
- [323] Z. Wu, C. Baek, C. You, and Y. Ma, “Incremental learning via rate reduction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1125–1133.
- [324] K. H. R. Chan, Y. Yu, C. You, H. Qi, J. Wright, and Y. Ma, “Redunet: A white-box deep network from the principle of maximizing rate reduction,” *arXiv preprint arXiv:2105.10446*, 2021.