Imperial College London

Imperial College London
Department of Computing

Learning Strategies for Improving Neural Networks for Image Segmentation under Class Imbalance

Zeju Li

Supervisor

Prof. Ben Glocker

Submitted in part fulfilment of the requirements for the degree of *Doctor of Philosophy* in Computing of Imperial College London

December 2022

Declaration

I, Zeju Li, declare that the work presented in this thesis in my own, unless otherwise acknowledged.

The formatting for this thesis is taken from the Thesis Template by R. Robinson at https://github.com/mlnotebook/thesis_template.

Copyright

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC).

Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes.

Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Acknowledgements

I feel very grateful for all the good time I had and meaningful things I took part in after coming alone 8,136 km away from home. I know the main credit for that goes to my supervisor Prof. Ben Glocker, to whom I want to express my sincerest gratitude. Thanks for offering me such a wonderful opportunity to pursue a PhD. His broad knowledge and invaluable guidance help me catch up with cutting-edge technologies. His kindness and trust make me confident and brave to explore what I think is important. Thanks for providing everything I need and teaching me how to be a perfect PhD supervisor.

I also want to thank Prof. Daniel Rueckert, Dr. Bernhard Kainz and Dr. Wenjia Bai for encouraging me through the PhD application process. Especially thank my second supervisor Prof. Daniel Rueckert for being an outstanding researcher and leader, making BioMedIA group a great place to do research.

I would like to thank all my collaborators in BioMedIA group. Most of all, I would like to thank my close collaborator and great mentor Konstantinos Kamnitsas, with whom I feel very lucky to work together. He always sets high standards for our research works and inspire me by patiently making things crystal clear. I would like to thank other collaborators Mobarak Islam, Qi Dou, Chen Chen, Cheng Ouyang and Chen Qin. I cannot realize most contributions without their supports. I am also grateful for the help and company from other colleagues and friends in BioMedIA group, which make this journey more pleasant and enjoyable! I would also like to thank Dr. Juan Eugenio Iglesias and Dr. Stamatia Giannarou to be my thesis examiners, who provide valuable discussions and detailed suggestions to make this thesis better. It is a great honor for me to have you contribute to my thesis.

I am thankful to my internship supervisors Prof. S. Kevin Zhou and Prof. Hu Han in Institute of Computing Technology before I formally enrolled in the PhD program. Their detailed guidance and timely support helped me get well prepared for the research afterwards. I also appreciate my internship supervisors Prof. Greg Slabaugh and Dr. Liang Chen in Huawei for providing the opportunity to apply machine learning algorithms in real-world applications.

At last, I would like to thank family for supporting me throughout my PhD study, especially during the last two years of separation because of COVID. Much appreciations to my parents for unconditional love and endless encouragements.

Abstract

This thesis aims to improve convolutional neural networks (CNNs) for image segmentation under class imbalance, which is referred to the problem of training dataset when the class distributions are unequal. We particularly focus on medical image segmentation because of its imbalanced nature and clinical importance.

Based on our observations of model behaviour, we argue that CNNs cannot generalize well on imbalanced segmentation tasks, mainly because of two counterintuitive reasons. CNNs are prone to overfit the under-represented foreground classes as it would memorize the regions of interest (ROIs) in the training data because they are so rare. Besides, CNNs could underfit the heterogenous background classes as it is difficult to learn from the samples with diverse and complex characteristics. Those behaviours of CNNs are not limited to specific loss functions.

To address those limitations, firstly we propose novel asymmetric variants of popular loss functions and regularization techniques, which are explicitly designed to increase the variance of foreground samples to counter overfitting under class imbalance. Secondly we propose context label learning (CoLab) to tackle background underfitting by automatically decomposing the background class into several subclasses. This is achieved by optimizing an auxiliary task generator to generate context labels such that the main network will produce good ROIs segmentation performance. Then we propose a meta-learning based automatic data augmentation framework which builds a balance of foreground and background samples to alleviate class imbalance. Specifically, we learn class-specific training-time data augmentation (TRA) and jointly optimize TRA and test-time data augmentation (TEA) effectively aligning training and test data distribution for better generalization. Finally, we explore how to estimate model performance under domain shifts when trained with imbalanced dataset. We propose class-specific variants of existing confidence-based model evaluation methods which adapts separate parameters per class, enabling class-wise calibration to reduce model bias towards the minority classes.

Contents

1 Introduction **1**

1.1 Motivation . . . 1

1.2 Objectives, Thesis Outline, Contributions and Publications . . . 4

2 Background **9**

2.1 Neural Networks . . . 10

2.1.1 Fundamentals . . . 10

2.1.2 Generalization in Deep Learning . . . 12

2.1.3 Meta-Learning . . . 15

2.2 Regularization in Deep Learning . . . 18

2.2.1 Task-Independent Data Augmentation . . . 19

2.2.2 Task-Driven Data Augmentation . . . 21

2.2.3 Other Regularization Approaches . . . 24

2.3 Image Segmentation . . . 26

2.3.1 Medical Image Segmentation . . . 27

2.3.2 Segmentation with Neural Networks . . . 28

2.4 Data Limitations in Medical Image Segmentation . . . 32

2.4.1 Class Imbalance . . . 32

2.4.2 Domain Shifts . . . 35

3 Overfitting under Class Imbalance **37**

3.1 Introduction . . . 38

3.2 Related Works . . . 39

3.2.1 Class Imbalance . . . 39

3.2.2 Regularization Techniques . . . 39

3.3 Analysis . . . 40

3.3.1 Under-Segmentation due to Class Imbalance . . . 40

3.3.2 Logit Distribution Shift . . . 42

3.4 Method . . . 45

3.4.1 Asymmetric Large Margin Loss . . . 46

3.4.2 Asymmetric Focal Loss . . . 47

3.4.3 Asymmetric Adversarial Training . . . 48

3.4.4 Asymmetric Mixup . . . 49

3.4.5 Asymmetric Data Augmentation . . . 50
3.4.6 The Combination of Asymmetric Techniques . . . 50
3.4.7 The Analysis of Large Margin Loss from a Sample Re-weighting Perspective 51
3.5 Experiments . . . 54
3.5.1 Experimental Setup . . . 54
3.5.2 Quantitative Results . . . 57
3.5.3 Logit Distribution Changes . . . 59
3.5.4 Sensitivity Analysis . . . 61
3.5.5 The Intensity Histograms of Different Datasets . . . 62
3.5.6 Quantitative Results of Abdominal Organ Segmentation . . . 63
3.6 Conclusion . . . 65

4 Underfitting under Class Imbalance 67
4.1 Introduction . . . 68
4.2 Related Works . . . 71
4.2.1 Class Imbalance . . . 71
4.2.2 Multi-task Learning . . . 71
4.3 Analysis . . . 72
4.3.1 Preliminaries . . . 72
4.3.2 False Positives on Background Class . . . 73
4.3.3 Underfitting of Background Samples . . . 75
4.3.4 The Effect of Context Labels . . . 77
4.4 Method . . . 78
4.4.1 Label Aggregation . . . 78
4.4.2 Context Constraints . . . 79
4.4.3 Task Generator Optimization with Meta-Gradients . . . 81
4.4.4 Implementation Details . . . 82
4.5 Experiments . . . 84
4.5.1 Experimental Setup . . . 84
4.5.2 Compared Methods and Processing . . . 85
4.5.3 Quantitative Results . . . 86
4.5.4 Results with Post-processing . . . 90
4.5.5 Effect on Logit Distributions . . . 90
4.5.6 Intensity Histograms of ROIs and Context Labels . . . 90
4.6 Conclusion . . . 96

5 Joint Optimization of Class-Specific Training- and Test-Time Data Augmentation 97
5.1 Introduction . . . 98
5.2 Related Works . . . 101

5.2.1 Data Augmentation Model . . . 101
5.2.2 Learning Based Training-Time Data Augmentation . . . 101
5.2.3 Learning Based Test-Time Data Augmentation . . . 102
5.3 Method . . . 103
5.3.1 Preliminaries . . . 103
5.3.2 Sampling Transformations . . . 103
5.3.3 Learning Scheme . . . 104
5.3.4 Overview of the Training Process . . . 105
5.3.5 Learning of Class-Specific Training-Time Data Augmentation . . . 106
5.3.6 Learning of Test-Time Data Augmentation . . . 109
5.3.7 Inference of Test-Time Data Augmentation . . . 110
5.3.8 Joint Learning of Training- and Test-Time Data Augmentation . . . 110
5.3.9 List of Operations for Training-Time Data Augmentation . . . 111
5.3.10 List of Transformation for Test-Time Data Augmentation . . . 113
5.3.11 Implementation Details . . . 114
5.4 Experiments . . . 117
5.4.1 Experimental Setup . . . 117
5.4.2 Compared Methods . . . 118
5.4.3 Quantitative Results . . . 119
5.4.4 Learned Policies with Different Settings . . . 124
5.4.5 Logit Distribution Maps . . . 129
5.4.6 Visualization of Validation Loss . . . 130
5.5 Conclusion . . . 133

6 Estimating Performance under Domain Shifts with Class-Specific Confidence Scores 134
6.1 Introduction . . . 135
6.2 Related Works . . . 137
6.3 Method . . . 138
6.3.1 Problem Setup . . . 138
6.3.2 A Unified Test Accuracy Framework Based on Model Calibration . . . 138
6.3.3 Class-Specific Temperature Scaling . . . 139
6.3.4 Class-Specific Difference of Confidences . . . 140
6.3.5 Class-Specific Average Thresholded Confidence . . . 140
6.3.6 Extension to Segmentation . . . 141
6.4 Experiments . . . 142
6.4.1 Setting . . . 142
6.4.2 Datasets . . . 143
6.4.3 Results and Discussion . . . 144
6.5 Conclusion . . . 148

7 Conclusions **149**
7.1 Summary of Contributions 150
7.2 Open Challenges and Future Outlook 153
7.2.1 Class Imbalance in Other 3D Vision Tasks 153
7.2.2 Class Imbalance in Self-Supervised Learning 154
7.2.3 Class Imbalance in Multi-Modality Learning 154
7.3 Conclusions 155

References **156**

List of Figures

1.1 Thesis motivations. . . . 2
1.2 Methodological contributions. . . . 4

2.1 The illustration of a multilayer perceptron (MLP). . . . 11
2.2 The schematic diagrams of good fitting, overfitting and underfitting for the classification problem. . . . 13
2.3 The generalization gaps when increasing model complexity of different machine learning models. . . . 14
2.4 The illustration of meta-learning. . . . 16
2.5 An overview of different data augmentation methods for image processing. . . . 18
2.6 The comparison of different prediction-driven model based data augmentation methods. . . . 22
2.7 An example of medical image segmentation. . . . 26
2.8 The illustration of a convolutional layer in CNN processing two-dimensional (2D) inputs. . . . 28
2.9 Two segmentation network architectures with successive convolutional layers. . . . 31
2.10 The illustration of class imbalance in medical image segmentation. . . . 32
2.11 The illustration of domain shifts in medical imaging. . . . 35

3.1 Performance on brain tumor core, brain stroke lesion, small organs, kidney and kidney tumor segmentation with varying amounts of training data. . . . 40
3.2 Visualization of different datasets and segmentation results with different portions of training data. . . . 41
3.3 Logit distributions when processing brain tumor segmentation (BraTS) with DeepMedic. 42
3.4 Logit distributions when processing kidney tumor segmentation (KiTS) with three-dimensional (3D) U-Net. . . . 43
3.5 Logit distributions when processing anatomical tracings of lesions after stroke (ATLAS) with DeepMedic. . . . 44
3.6 The illustration of the proposed asymmetric modifications for the existing loss functions and regularization techniques. . . . 45
3.7 Activations of the classification layer when processing tumor (top) and background (bottom) samples of BraTS with DeepMedic, using 5% training data. . . . 60

3.8 Activations of the classification layer for tumor (top), kidney (middle) and background (bottom) samples of KiTS with 3D U-Net, using 10% training data. 61
3.9 The intensity histogram of different datasets. 63
3.10 The class distribution of the abdominal organ dataset we use in this study. . . . 64

4.1 Effects of context labels on the segmentation accuracy. 70
4.2 Visualization of different datasets and segmentation results on test data when training w/o and w/ different types of context labels. 72
4.3 The distribution of activations (logits) after the classification layer using ROI (blue) or background samples (orange) of test data as network input. 74
4.4 Effects of context labels on the learning of neural networks. 75
4.5 The distribution of activations (logits) after the classification layer using ROI (blue) or background samples (orange) of training data as network input. 76
4.6 The illustration of the training process of CoLab 78
4.7 The illustration of the label aggregation process. 79
4.8 The illustration of context constraints. 80
4.9 Examples of the results $\tilde{f}_\theta(\boldsymbol{x}_i)$ of training data with different types of context labels. 87
4.10 Additional examples of the results $\tilde{f}_\theta(\boldsymbol{x}_i)$ of training data with different types of context labels. 88
4.11 The intensity histogram of ROIs calculated based on $\boldsymbol{y}_i$ and the different context labels calculated based on $\tilde{f}_\theta(\boldsymbol{x}_i)$. 95

5.1 Data augmentation improves model performance by aligning the training and validation/test data distribution. 100
5.2 The optimization process of the proposed method. 103
5.3 The learning scheme of the proposed method. 105
5.4 The illustration of one iteration during the training process. 111
5.5 Visualization of K=84 choices of transformations for TEA we use in this study. . 113
5.6 The sampling distribution for three heuristic TRA policies. 119
5.7 The heuristic data augmentation policy and the learned probability distributions over augmentations based on different segmentation models for brain stroke lesion segmentation with 100% ATLAS training data. 124
5.8 Visualization of different datasets and segmentation results with different data augmentation methods. 125
5.9 The learned probability distributions over augmentations based on different segmentation models for brain stroke lesion segmentation with 50% ATLAS training data. 126

5.10 The learned probability distributions over augmentations based on different segmentation models for kidney and kidney tumor segmentation with 50% KiTS training data. . . . 127
5.11 The learned probability distributions over augmentations based on different segmentation models for kidney and kidney tumor segmentation with 100% KiTS training data. . . . 128
5.12 The learned probability distributions over augmentations based on different segmentation models for abdominal organ segmentation. . . . 129
5.13 The learned probability distributions over augmentations based on different segmentation models for cross-site prostate segmentation. . . . 130
5.14 The logit distribution learned with different data augmentation policies. . . . 131
5.15 Validation loss curves of different models trained for kidney and kidney tumor segmentation with KiTS. . . . 132
5.16 Validation loss curves of different models trained for brain lesion segmentation with ATLAS. . . . 132

6.1 Effect of class imbalance on confidence-based model evaluation methods. . . . 135
6.2 The illustration of proposed Class-Specific modifications for four existing model evaluation methods. . . . 139
6.3 Real versus predicted accuracy/Dice similarity coefficient (DSC) for classification/segmentation tasks respectively. . . . 142
6.4 Visualization of the datasets containing domain shifts. . . . 143
6.5 Boxplots of the mean absolute error (MAE) results for different methods. . . . 147

List of Tables

2.1 The statistics and class imbalance ratios of the four datasets used in this study. . 33

3.1 Evaluation of brain tumor core segmentation using DeepMedic with different amounts of training data and different techniques to counter overfitting. 54
3.2 Evaluation of brain stroke lesion segmentation on ATLAS based on DeepMedic with different amounts of training data and different techniques to counter overfitting. 55
3.3 Evaluation of kidney and kidney tumor segmentation based on 3D U-Net with different amounts of training data and different techniques to counter overfitting. . . 56
3.4 The sensitivity analysis of different hyper-parameters. 62
3.5 Evaluation of abdominal organ segmentation with 25% of training data with symmetric (sy.) and asymmetric (asy.) focal loss. 65

4.1 Evaluation of different binary segmentation tasks with different types of context label generation approaches. 92
4.2 Evaluation of different binary segmentation tasks with different types of context labels approaches with post-processing. 93
4.3 Evaluation of pancreas and pancreatic tumor mass segmentation with different types of context label generation approaches. 94
4.4 Evaluation of pancreas and pancreatic tumor mass segmentation with different types of context label generation approaches with post-processing. 94

5.1 List of all operations that our method can choose for TRA. 112
5.2 List of all operations that our method can choose for TEA. 116
5.3 Evaluation of brain stroke lesion segmentation on ATLAS based on different network architectures with different amounts of training data using different data augmentation methods. 120
5.4 Evaluation of kidney and kidney tumor segmentation based on different network architectures with different amounts of training data using different data augmentation methods. 121
5.5 Evaluation of abdominal organ segmentation based on different network architectures using different data augmentation methods. 122
5.6 Evaluation of cross-site prostate segmentation based on different network architectures using different data augmentation methods. 123

6.1 Dataset statistics of skin lesion classification datasets. 144
6.2 Evaluation on different tasks under varied types of domain shifts based on MAE. . 145
6.3 Evaluation on different tasks under varied types of domain shifts based on R2 Score. 146

Acronyms

2D two-dimensional. xii, 28, 29, 32, 153
3D three-dimensional. xii, xiii, xv, 2, 26, 28–32, 40, 43, 56, 61, 74, 84, 110, 117, 121–123, 142, 153

AC average confidence. 135, 137, 138, 142, 145, 146
akiec actinic keratoses. 143, 144
ANN artificial neural network. 9
AO aorta. 118, 122
ATC average thresholded confidence. 134–136, 138, 140, 141, 145, 146, 152
ATLAS anatomical tracings of lesions after stroke. xii–xv, 33, 40, 44, 55, 57, 59, 62, 63, 85, 117, 119, 120, 122, 124–126, 129, 132, 144

bcc basal cell carcinoma. 143, 144
BCE binary cross entropy. 81, 84
BG background. 33, 40, 124
bkl nenign keratosis. 143, 144
BraTS brain tumor segmentation. xii, 26, 33, 40, 42, 54, 57, 59–63

CE cross entropy. 40, 41, 46, 48, 54, 55, 57, 62, 65, 73, 84, 117, 130
CNN convolutional neural network. vi, xii, 9, 28–32, 35, 40, 42, 43, 45, 46, 51, 68, 72–78, 151
CoLab context label learning. vi, xiii, 4, 5, 67–72, 74, 76–78, 81–94, 96, 105, 151
CT computerised tomography. 1, 26, 30, 55, 57, 73, 74, 84–88, 90, 117, 127, 128

df dermatofibroma. 143, 144
DNN deep neural network. 1–4, 9–15, 27, 34–36
DoC difference of confidences. 134–136, 138, 140, 141, 145, 146, 152
DSC Dice similarity coefficient. xiv, 34, 40, 41, 46–48, 51, 54–57, 59, 62, 64, 65, 68, 70, 71, 73, 77, 81, 84, 86, 89, 117, 119, 122, 123, 130, 134, 136, 141, 142, 144, 145, 148

E esophagus. 117, 122
ECE Expected Calibration Error. 148
ERM empirical risk minimization. 11

FG foreground. 33, 40, 124
FLAIR fluid-attenuated inversion recovery. 59
FN false negative. 42, 46
FP false positive. 46, 68, 71, 72, 74, 76, 77, 86, 88–90

GAN generative adversarial network. 20–22
GB gallbladder. 117, 122
GELU gaussian error linear unit. 11
GPU graphics processing unit. 11

HD 95% Hausdorff distance. 57, 68, 70, 86, 119
HOG Histogram of Oriented Gradients. 12

HU Hounsfield units. 84, 117

IID independently and identically distributed. 135
IVC inferior vena cava. 118, 122

KiTS kidney tumor segmentation. xii–xiv, 33, 40, 41, 43, 57, 59–63, 84, 86, 113, 117, 123, 127–129, 132
KL Kullback–Leibler. 22, 24

LA left adrenal gland. 118, 122
LiTS liver tumor segmentation. 84, 86
LIV liver. 118
LK left kidney. 117
LMLE large margin local embedding. 39

MAE mean absolute error. xiv, xvi, 142, 144, 145, 147
MALP-EM expectation–maximisation based refinement. 77, 85
mel melanoma. 143, 144
MLP multilayer perceptron. xii, 11, 28, 29
MR magnetic resonance. 1, 3, 26, 35, 54, 55, 73, 77, 84, 85, 87, 117, 118, 144
MRF Markov random field. 27, 90
MRI magnetic resonance imaging. 1, 26, 30, 35, 36, 86, 118, 129, 144
MTL multi-task learning. 71

NORCAL normalized calibration. 136, 142, 145, 146
nv melanocytic nevi. 143, 144

PA pancreas. 118, 122
PRC precision. 41, 57, 58, 77, 86, 119

QC quality control. 35, 142, 145, 146

RA right adrenal gland. 118, 122
ReLU rectified Linear Unit. 11
RK right kidney. 117
ROC receiver operating characteristic. 58
ROI region of interest. vi, xiii, 1–4, 20, 26, 32, 33, 35, 37, 67, 68, 70–81, 84–91, 95, 96, 148, 150, 151

SEN sensitivity. 41, 57, 58, 77, 86, 119
SGD stochastic gradient descent. 12
SOTA state-of-the-art. 5, 10, 21, 23, 27, 84, 97, 134, 136, 148
SP spleen. 117
SSL self-supervised learning. 154
STO stomach. 118

TEA test-time data augmentation. vi, xiii, xv, 5, 97–100, 102, 104–106, 109, 110, 113, 116, 118, 119, 122–124, 126, 128, 133, 151
TP true positive. 42, 46, 68
TRA training-time data augmentation. vi, xiii, xv, 5, 97–102, 104–108, 110–112, 118–122, 124–130, 133, 151

TS temperature scaling. 134–136, 138, 139, 141–143, 145, 146, 152

V portal vein and splenic vein. 118, 122
VAE variational autoencoder. 20
vasc vascular. 143, 144
VC Vapnik–Chervonenkis. 15
VS vestibular schwannoma. 73, 85, 88, 89, 136, 142, 145, 146

Chapter 1

Introduction

1.1 Motivation

Machine learning algorithms, which are optimized to learn the mapping between input and output factors with varied types of models, can discover hidden pattern from data implicitly. Neural networks, as one of the most prevalent machine learning models, are inspired by biological neural networks and consist of a group of connected neurons. A deep neural network (DNN) is a neural network with multiple layers and can be utilized to solve many real world learning tasks including speech recognition [59, 100], natural language processing [58, 225] and computer vision [48, 135]. Specifically, DNNs provide the level of performance comparable to human expert for applications such as recognizing human faces [56], playing the board game Go [217] and interpreting radiographs [214].

Semantic segmentation is one of the computer vision tasks, requiring to partition an image into multiple image segments with semantic meanings [176]. As one of the semantic segmentation tasks, medical image segmentation is the process of delineating the contour of regions of interest (ROIs) in medical images such as magnetic resonance (MR) or computerised tomography (CT) images. Medical image segmentation aims to isolate specific anatomical regions from the background for subsequent analysis with clinical or research purposes, such as volumetric and shape analysis [206]. For example, we show the example of brain lesion segmentation from T1 weighted magnetic resonance imaging (MRI) in Fig. 1.1(a). Specifically, to segment brain lesion can facilitate treatment planning, improve diagnostics and growth rate prediction [152]. However, the segmentation process is time-consuming and needs reference to expert knowledge. Therefore, automatic segmentation is favorable for clinical applications and has been one of the most important tasks for medical image computing.

Recently, deep learning has shown promising results in automatic image segmentation in terms of both accuracy and computational speed [44, 165]. Deep learning approaches learn features directly from the training data with a supervised learning paradigm, where experts have annotated the semantic content. However, its success heavily depends on the quality and amount of labelled training data from which the models can learn. In practice, a large amount of high-quality data for medical image segmentation is hard to obtain. Therefore, the under-represented training data is the bottleneck of model performance. Specifically, we find class imbalance and domain shifts in

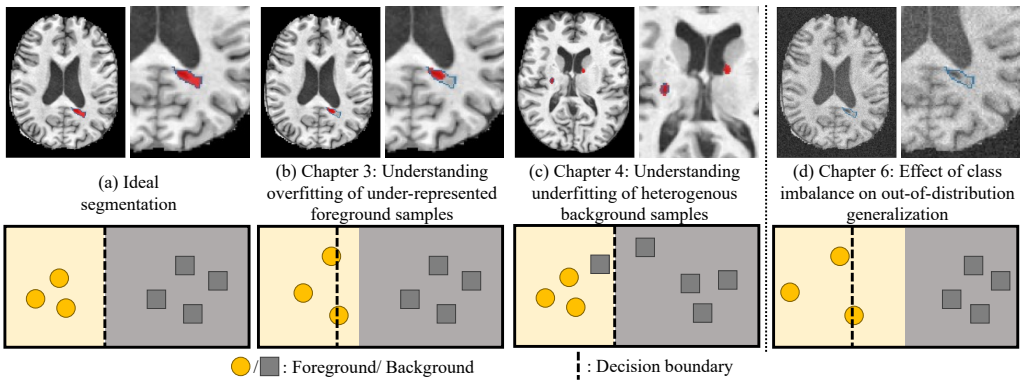


Figure 1.1: Thesis motivations. The summary of our observations on neural networks based image segmentation under class imbalance. (Top part) We show the ground truth manual labels for brain lesion in red regions, predictions in blue boundaries. (Bottom part) We illustrate the corresponding learned logits (activations of the classification layer). We find that when trained for small lesion segmentation, (b) neural networks tend to systemically under-segment the ROIs because of overfitting of foreground samples and (c) over-segment the targets as a result of underfitting of background samples. (d) We also observe that neural networks are prone to under-segment the small ROIs with the existence of domain shifts.

medical imaging would always impede deep learning models to generalize well. Motivated by this, in this work we analyze network behaviours under such conditions and propose several learning strategies to improve the robustness and reliability of deep learning models.

Class imbalance in nature is a common issue for medical image segmentation as medical imaging are mostly three-dimensional (3D) scans while ROIs are relatively small compared with the large volume of background voxels [111, 115]. For example, as shown in Fig. 1.1(a), the size of brain lesion region in MR images is less than 1/1,000 the size of the image volume. Under such imbalanced distribution, we find that a DNN cannot always generalize well mainly because of two reasons.

- Firstly, instead of extracting generalizable features, DNN would just memorize the small ROIs in the training data because they are so rare. In that case, which is referred as overfitting, the class bias and the limited amount of training data would lead to a systematic under-segmentation of small structures on unseen test data, as shown in Fig. 1.1(b).

- Secondly, we find that DNN cannot fit the background training data well because of the large intra-class variance of heterogeneous background samples. As a result this underfitting problem, DNN would fail to identify the background samples sharing similar characteristics with the ROIs. Consequently, the models would produce false positive predictions, leading to systematic over-segmentation, as illustrated in Fig. 1.1(c).

As in practice the segmentation is affected by the interaction of both factors, few previous studies

explain and analyze these problems in a clear way. More importantly, there is a need for advanced DNN learning strategies to improve small ROI segmentation accuracy by combating foreground overfitting and background underfitting.

A domain shift refers to the setting in which a machine learning model is deployed and trained with different data distributions [14]. As a data-driven approach, current DNN could perform poorly under domain shifts in an unexpected way. Specifically, models trained with data from the source domain always cannot generalize well on data from the target domain if domains vary greatly. For example, neural network would fail to segment brain lesion with noisy MR images, as shown in Fig. 1.1(d). In practice, domain shifts in medical imaging could due to different acquisition settings, different imaging contrast, different annotations or/and different populations [33]. Specifically, we find domain shifts are always entangled with class imbalance for medical image segmentation and neural networks tend to under-perform the segmentation of the under-represented classes. Detecting such failures automatically at the deployment stage is crucial for safety-critical applications such as medical imaging.

Data augmentation, as a common regularization technique for machine learning models, can improve the generalization capacity of DNN by extending the training dataset with artificial samples [216, 233]. The data augmentation policy would benefit network training if the augmented samples show characteristics which is similar to that of data it encounters when deployed but has not caught with the original training dataset. As anatomy heterogeneity and imaging quality differ across datasets, optimal data augmentation policy varies considerably. Designing a set of effective data augmentation policy is a crucial but laborious task for medical imaging practioners.

1.2 Objectives, Thesis Outline, Contributions and Publications

Taking the factors above into consideration, in this thesis, we focus on analyzing the effects of class imbalance on neural networks for image segmentation and advancing the learning of neural networks for robust and reliable generalization. Specifically, we have three complementary goals:

- To investigate the effects of class imbalance on model behaviours for image segmentation. In order to gain a deep understanding of model generalization and identify the limitation of DNNs under class imbalance, we monitor the learned network logit distributions with extensive systematic experiments.
- To advance learning strategies for neural networks to improve segmentation accuracy under class imbalance. Beyond empirical analysis, we aim to improve neural networks for small ROI segmentation with specific learning strategies including novel loss functions, regularization techniques and multi-task learning.
- To develop performance estimation algorithms to facilitate safe deployment of DNNs in real-world settings. We seek to detect the failure of machine learning models under the setting of domain shifts. In particular, we take class imbalance into account and calibrate model confidence to match class-wise accuracy.

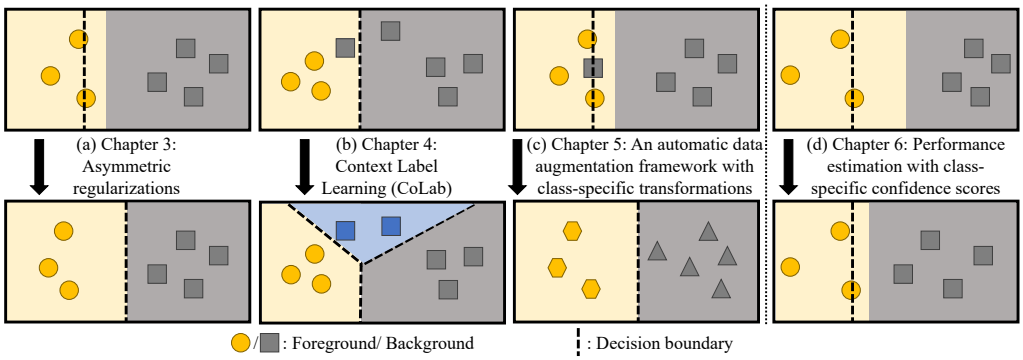


Figure 1.2: Methodological contributions. The summary of the main methodological contributions of this thesis. (a) We propose a series of asymmetric loss functions and regularization techniques to alleviate the overfitting of foreground samples. (b) We propose context label learning (CoLab) for background class decomposition to reduce the underfitting of background samples. (c) We propose an automatic data augmentation framework with class-specific transformations, building a balance of foreground and background samples to mitigate the class imbalance issues. (d) We propose to utilize class-wise calibration within the framework of confidence based performance estimation, reducing the biases induced by class imbalance.

The remainder of this thesis is organized as below.

Chapter 2 introduces the basic concepts of neural networks, with a focus on deep learning and its applications on computer vision. It also highlights the class imbalance and domain shifts problems in image segmentation. This background information is necessary for us to clarify our motivations and back up our arguments.

Chapter 3 analyzes and alleviates the overfitting of under-represented foreground samples for image segmentation. We find empirically that when training with limited data and strong class imbalance, at test time the distribution of logit activations may shift across the decision boundary, while samples of the well-represented class seem unaffected. This bias leads to a systematic under-segmentation of small structures, as shown in Fig. 1.1(b). Hereby, we introduce new asymmetric variants of popular loss functions and regularization techniques which are explicitly designed to counter logit shift of the under-represented classes as illustrated in Fig. 1.2(a).

Chapter 4 investigates and tackles the underfitting of heterogeneous background samples for image segmentation. We find the distribution over background logit activations may also shift across the decision boundary even with sufficient training data, leading to systematic over-segmentation, as shown in Fig. 1.1(c). Motivated by the observations that human-defined context labels would benefit model training, we propose CoLab to improve the context representations by decomposing the background class into several subclasses, as illustrated in Fig. 1.2(b). Specifically, we train an auxiliary network as a task generator, along with the primary segmentation model, to automatically generate context labels that positively affect the segmentation accuracy.

Chapter 5 introduces a general automatic data augmentation framework for image segmentation, with a special concern about the class imbalance problem. Specifically, we propose to adopt different kinds of training-time data augmentation (TRA) for training samples from different classes, effectively extending the training data distribution and alleviating the class imbalance issue by adaptively balancing the variances of augmented foreground and background samples. We demonstrate this idea in Fig. 1.2(c). In addition, we jointly optimize TRA and test-time data augmentation (TEA), which are closely connected as both aim to align the training and test data distribution but were so far considered separately in previous works.

Chapter 6 focuses on identifying performance degradation under domain shifts without the ground truth. Model confidence of predictions on the unlabeled test data can be utilized to estimate the model performance. Existing methods calibrate model confidence based on the data from the source domain to achieve high estimation accuracy. However, we find these methods would be always under confident about the minority classes as a result of class imbalance. This is because the class bias would make the decision boundary systematically shift toward the minority classes under domain shifts, as shown in Fig. 1.1(d). Therefore, we derive class-specific modifications of state-of-the-art (SOTA) confidence-based model evaluation methods which adapt separate parameters per class, enabling class-wise calibration. We illustrate this idea in Fig. 1.2(d).

Chapter 7 summarizes the main findings and contributions of this thesis. We look into other machine learning research problems and highlight some opening challenges which are related to our investigations on class imbalance.

This thesis is based on the following research publications:

1. **Z. Li**, K. Kamnitsas and B. Glocker. Overfitting of Neural Nets Under Class Imbalance: Analysis and Improvements for Segmentation. International Conference on Medical Image Computing and Computer-Assisted Intervention, 2019: 402-410. [146]

2. **Z. Li**, K. Kamnitsas and B. Glocker. Analyzing Overfitting Under Class Imbalance in Neural Networks for Image Segmentation. IEEE Transactions on Medical Imaging, 2020, 40(3): 1065-1077. [147]

3. **Z. Li**, K. Kamnitsas, C. Ouyang, C. Chen and B. Glocker. Context Label Learning: Improving Background Class Representations in Semantic Segmentation. IEEE Transactions on Medical Imaging, 2022, Provisionally Accepted. [149]

4. **Z. Li**, K. Kamnitsas, Q. Dou, C. Qin and B. Glocker. Joint Optimization of Class-Specific Training- and Test-Time Data Augmentation in Segmentation. resubmitted to IEEE Transactions on Medical Imaging.

5. **Z. Li**, K. Kamnitsas, M. Islam, C. Chen and B. Glocker. Estimating Model Performance under Domain Shifts with Class-Specific Confidence Scores. International Conference on Medical Image Computing and Computer-Assisted Intervention, 2022: 693-703. [148]

During my PhD study, I also got involved in other research projects. Although those works are not included in this thesis, they are all closely related to the research topics. Here follows the complete list of published works:

1. **Z. Li**, H. Li, H. Han, G. Shi, J. Wang and S. K. Zhou. Encoding CT Anatomy Knowledge for Unpaired Chest X-ray Image Decomposition. International Conference on Medical Image Computing and Computer-Assisted Intervention, 2019: 275-283. [150]

2. H. Li, H. Han, **Z. Li**, L. Wang, Z. Wu, J. Lu and S. K. Zhou. High-Resolution Chest X-Ray Bone Suppression Using Unpaired CT Structural Priors. IEEE Transactions on Medical Imaging, 2020, 39(10): 3053-3063. [139]

3. H. Luo, Q. Zhuang, Y. Wang, A. Abudumijiti, K. Shi, A. Rominger, H. Chen, Z. Yang, V. Tran, G. Wu, **Z. Li**, Z. Fan, Z. Qi, Y. Guo, J. Yu and Z. Shi. A Novel Image Signature-Based Radiomics Method to Achieve Precise Diagnosis and Prognostic Stratification of Gliomas. Laboratory Investigation, 2021, 101: 450-462. [166]

4. Q. Dou, T. Y. So, M. Jiang, Q. Liu, V. Vardhanabhuti, G. Kaissis, **Z. Li**, W. Si, H. H. C. Lee, K. Yu, Z. Feng, L. Dong, E. Burian, F. Jungmann, R. Braren, M. Makowski, B. Kainz, D. Rueckert, B. Glocker, S. C. H. Yu and P. A. Heng. Federated Deep Learning for Detecting COVID-19 Lung Abnormalities in CT: A Privacy-Preserving Multinational Validation Study. npj Digital Medicine, 2021, 4: 60. [64]

5. M. Qiao, C. Liu, **Z. Li**, J. Zhou, Q. Xiao, S. Zhou, C. Chang, Y. Gu, Y. Guo and Y. Wang. Breast Tumor Classification Based on MRI-US Images by Disentangling Modality Features. IEEE Journal of Biomedical and Health Informatics, 2021, 26(7): 3059-3067. [197]

6. C. Chen, **Z. Li**, C. Ouyang, M. Sinclair, W. Bai and D. Rueckert. Maxstyle: Adversarial Style Composition for Robust Medical Image Segmentation. International Conference on Medical Image Computing and Computer-Assisted Intervention, 2022: 151-161. [39]

7. X. Gu, Y. Guo, **Z. Li**, J. Qiu, Q. Dou, Y. Liu, B. Lo and G. Z. Yang. Tackling Long-Tailed Category Distribution under Domain Shifts. European Conference on Computer Vision. 2022: 727-743. [81]

8. C. Chen, C. Qin, C. Ouyang, **Z. Li**, S. Wang, H. Qiu, L. Chen, G. Tarroni, W. Bai and D. Rueckert. Enhancing MR Image Segmentation with Realistic Adversarial Data Augmentation. Medical Image Analysis, 2022, accepted. [40]

9. C. Ouyang, S. Wang, C. Chen, **Z. Li**, W. Bai, B. Kainz and D. Rueckert. Improved Post-Hoc Probability Calibration for Artifact-Corrupted MRI Segmentation. Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, 2022: 59-69 [188]

10. L. Li, Q. Ma, **Z. Li**, C. Ouyang, W. Zhang, A. Price, V. Kyriakopoulou, L. Cordero-Grande, A. Makropoulos, J. Hajnal, D. Rueckert, B. Kainz and A. Alansary. Fetal Cortex Segmentation with Topology and Thickness Loss Constraints. Topological Data Analysis and Its Applications for Medical Data, 2022: 123-133 [141]

11. A. Hering, L. Hansen, T. C. W. Mok, A. C. S. Chung, H. Siebert, S. Hager, A. Lange, S. Kuckertz, S. Heldmann, W. Shao, S. Svesal, M. Rusu, G. A. Sonn, T. Estienne, M. Vakalopoulou, L. Han, Y. Huang, P. Yap, M. Brudfors, Y. Balbastre, S. Joutard, M. Modat, G. Lifshitz, D. Rivav, J. Lv, Q. Li, V. Jaouen, D. Visvikis, C. Fourcade, M. Rubeaux, W. Pan, Z. Xu, B. Jian, F. D. Benetti, M. Wodzinski, N. Gunnarsson, J. Sjolund, D. Grzech, H. Qiu, **Z. Li**, A. Thorley, J. Duan, C. Großbrohmer, A. Hoopes, I. Reinertsen, Y. Xiao, B. Landman, Y. Huo, K. Murphy, N. Lessmann, B. V. Ginneken, A. V. Dalca, M. P. Heinrich. Learn2Reg: comprehensive multi-task medical image registration challenge, dataset and evaluation in the era of deep learning. IEEE Transactions on Medical Imaging, 2022: accepted [99]

12. C. Ouyang, C. Chen, S. Li, **Z. Li**, C. Qin, W. Bai and D. Rueckert. Causality-inspired Single-

source Domain Generalization for Medical Image Segmentation. IEEE Transactions on Medical Imaging, 2022: accepted [187]

Chapter 2

Background

This chapter outlines some basic concepts of machine learning along with a few relevant technical details. We will then describe our contributions in the following chapters on the basis of these fundamental concepts.

Neural networks, also known as artificial neural networks (ANNs), are one of the key components of this thesis. Here we first introduce some fundamentals of neural networks, with a focus on the generalization capacity under different conditions. We also explain the concepts of meta-learning which is a learn strategy with high-level objectives.

Modern deep neural networks (DNNs) rely on regularization techniques to generalize well. Data augmentation is a de facto standard regularization for DNNs. We review varied data augmentation techniques for computer vision as they are related to some technical aspects of our contributions.

Image segmentation is the image analysis task we study in this thesis. We particularly introduce and focus on medical image segmentation due to its technological challenges and clinical importance. We describe convolutional neural networks (CNNs), a class of neural network designed for computer vision, and its usage in segmentation.

As a data-driven approach, the success of DNNs heavily depends on the quality of the data it trained with. We highlight class imbalance and domain shifts as two main data limitations that impede DNNs generalization for medical image segmentation.

2.1 Neural Networks

Neural networks are one of the probabilistic classifiers or regressors used in machine learning algorithms. DNNs, or deep learning, allow to make predictions with high speed and great accuracy. Neural networks have been successfully applied to a wide variety of applications such as computer vision [48, 135] and speech recognition [59, 100], where the probability of an outcome is predicted given a set amount of input data [94, 127]. This work focuses on improving state-of-the-art (SOTA) DNNs, particularly with a focus on the task of image segmentation. In this section, we will introduce some fundamentals of neural network and its advanced learning algorithms, as the basis of the subsequent works in this thesis.

2.1.1 Fundamentals

We consider a supervised learning paradigm and neural networks for the classification task with the data drawn from the distribution $\mathcal{D}$. Normally, we are given a training dataset $\mathcal{D}_{train} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$ with N samples, where $\boldsymbol{x}_i \in \mathbb{R}^{\mathrm{d}}$ is the observation such as an image or pattern and $\boldsymbol{y}_i$ is the desired outcome. $\boldsymbol{y}_i \in \mathbb{R}^{\mathrm{c}}$ is commonly encoded with the one-hot encoding vector such that its j'th element is corresponding to the j'th class such as $y_{ij} \in \{0, 1\}$ while $\mathbf{1} \cdot \boldsymbol{y}_i = 1 \, \forall \, i$. A classification model f, such as a neural network, is the function that maps the input samples to the label space. The parameters of f are learnt with the training dataset $\mathcal{D}_{train}$. The learned model is expected to be deployed to the unseen test dataset $\mathcal{D}_{test}$ with minor modifications. In addition to the training dataset $\mathcal{D}_{train}$ and the test dataset $\mathcal{D}_{test}$, typically we have another validation dataset which can help us choose hyper-parameters for model training.

Neuron is the basic predictive model of neural networks which is inspired by the neuron in a biological brain. We illustrate the structure of a neuron in Fig. 2.1(a). A neuron utilizes a set of weight $\boldsymbol{w}^o = [w_1^o, w_2^o, w_3^o]$ and bias b^o to connect two layers of nodes including an input layer $\boldsymbol{x}_i$ and an output layer h_i^o. Specifically, h_i^o is obtained as the linear combination of $\boldsymbol{x}_i$ and the network parameters including $\boldsymbol{w}^o$ and b^o:

$$h_i^o = \boldsymbol{w}^o \boldsymbol{x}_i + b^o = \sum_{j=1}^{3} w_j^o x_{ij} + b^o, \tag{2.1}$$

In order to model the non-linear relationship between the two nodes, an activation function σ is always adopted to transform h_i^o to the final output z_i^o:

$$z_i^o = \sigma(h_i^o), \tag{2.2}$$

The activation function is a core design for neural network. In practice, activation functions such

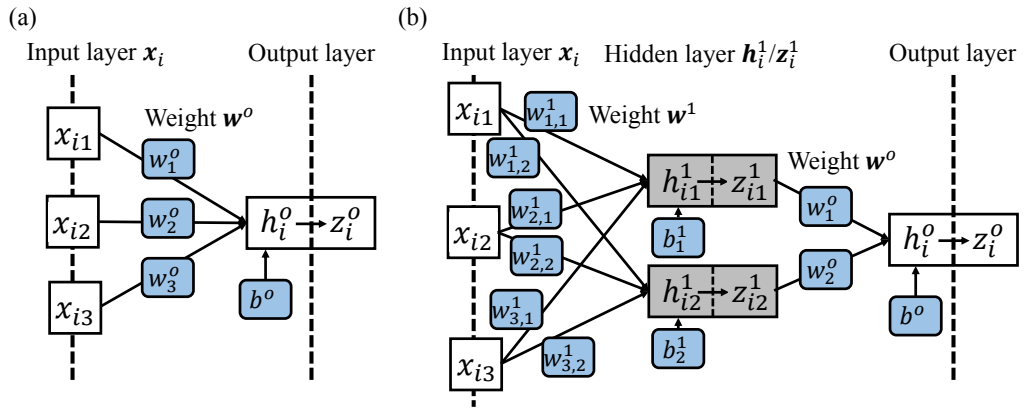


Figure 2.1: The illustration of a multilayer perceptron (MLP). (a) The structure of a neuron, which consists of two layers of nodes: an input layer $\mathbf{x}_i$ and an output layer h_i^o/z_i^o. The two layers are connected with weight $\mathbf{w}^o$ and bias b^o. The neuron uses an activation function to produce the output p_i^o given the input from the previous layer z_i^o. (b) The structure of an MLP with a single hidden layer $\mathbf{h}_i^1/\mathbf{z}_i^1$. Except for the input layer, each layer contains an activation function.

sigmoid, rectified Linear Unit (ReLU) [167], gaussian error linear unit (GELU) [98] and leaky ReLU [167] as are chosen for different applications.

An MLP, which is a simple version of neural network, is the combination of a series of stacked neurons. We illustrate a MLP with three layers of nodes in Fig. 2.1(b). Different from a neuron, a mlp contains at least a hidden layer between the input and output layer. Specifically, by calculating using the weight and bias of the hidden layer $\boldsymbol{w}^1 = \begin{bmatrix} w_{1,1}^1 & w_{2,1}^1 & w_{3,1}^1 \\ w_{1,2}^1 & w_{2,2}^1 & w_{3,2}^1 \end{bmatrix}$ and $\boldsymbol{b}^1 = [b_1^1, b_2^1]$, we can obtain:

$$h_i^o = \boldsymbol{w}^o(\boldsymbol{w}^1\boldsymbol{x}_i + \boldsymbol{b}^1) + b^o. \tag{2.3}$$

Similarly, more hidden layers can be added into the structure of MLP. DNNs refers to neural networks with multiple hidden layers [133]. The use of advanced specialized electric circuit such as graphics processing unit (GPU) makes it possible for the efficient training and inference of modern DNNs containing an enormous number of network parameters with hundreds of hidden layers [163].

The network parameters $\theta = (\boldsymbol{w}^1, \boldsymbol{b}^1, \boldsymbol{w}^o, b^o)$ are learned with the training dataset $\mathcal{D}_{train}$. Typically, we aim to minimize the expected risk $R_{\mathcal{L},\mathcal{D}}(f_\theta)$, which is measure of model performance on a known set of training data. However, the expected risk $R_{\mathcal{L},\mathcal{D}}(f_\theta)$ cannot be calculated as the real data distribution $\mathcal{D} \supset \mathcal{D}_{train}$ is usually inaccessible. Instead, θ is optimized based on empirical risk minimization (ERM). Specifically, given any cost function $\mathcal{L}$, the empirical risk of the model f_θ

is defined as $R_{\mathcal{L},\mathcal{D}_{train}}(f_\theta) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(f_\theta(\boldsymbol{x}_i), \boldsymbol{y}_i)$. We aim to calculate the optimal θ^* such that $\theta^* = \text{argmin}_\theta\, R_{\mathcal{L},\mathcal{D}_{train}}(f_\theta)$. The optimization process is always achieved with gradient descend based techniques for neural networks. Specifically, the parameter θ is updated step by step with the calculated gradient of the cost $\mathcal{L}$ with respect to the parameters:

$$\theta^{t+1} = \theta^t - \alpha\frac{\partial\mathcal{L}(f_\theta(\boldsymbol{x}_i), \boldsymbol{y}_i)}{\partial\theta^t}, \tag{2.4}$$

where α is the learning rate.

In practice, the gradients in neural networks are computed with backpropagation [134]. Specifically, in order to fit neural networks, the gradients on θ are calculated with the partial derivative of the cost function $\mathcal{L}$ using the chain rule. Compared with the direct computation of gradients, backpropagation is computationally efficient and memory efficient as the gradients are computed one by one from the last layer, avoiding the redundant computation and store of the immediate results of the chain rule. The efficiency makes it possible for the optimization of multi-layer networks.

For example, $\boldsymbol{w}^1$ can be updated with the gradient calculated as:

$$\frac{\partial\mathcal{L}(f_\theta(\boldsymbol{x}_i), \boldsymbol{y}_i)}{\partial\boldsymbol{w}^1} = \frac{\partial\mathcal{L}(f_\theta(\boldsymbol{x}_i), \boldsymbol{y}_i)}{\partial z_i^o}\frac{\partial z_i^o}{\partial h_i^o}\frac{\partial h_i^o}{\partial \boldsymbol{z}_i^1}\frac{\partial \boldsymbol{z}_i^1}{\partial \boldsymbol{h}_i^1}\frac{\partial \boldsymbol{h}_i^1}{\partial \boldsymbol{w}^1}. \tag{2.5}$$

In similar ways, all the gradient of the network parameters can calculated with low computation as the forward intermediate results are stored. After then optimization algorithms such as stochastic gradient descent (SGD) [202] and Adam [125] can be utilized to update the network parameters θ.

Therefore, DNNs is data-driven approaches which can learn hierarchical multi-layer representations directly from the training samples, in contrast to conventional machine learning approaches where features are designed based on prior knowledge such as Gaussian derivatives and the Histogram of Oriented Gradients (HOG) features [194, 266]. This is part of the reason why DNNs are more powerful as the learned features are more complex than hand-engineered features and especially tailored for the given tasks and data.

2.1.2 Generalization in Deep Learning

It is important to understand model generalization to identify and tackle limitations of DNNs that restrict the adaptation to complex applications. After the model f_θ is learned with the training dataset $\mathcal{D}_{train}$, we want it to perform well with the real data distribution $\mathcal{D}$. In other words, we aim to make expected risk $R_{\mathcal{L},\mathcal{D}}(f_\theta)$ as low as possible. As we cannot access the all the data in $\mathcal{D}$, practically we utilize a test dataset $\mathcal{D}_{test} \subset \mathcal{D}$, which is drawn from the same distribution but does not overlap with $\mathcal{D}_{train}$ to quantify the model performance. Then, the empirical risk on the

test data $R_{\mathcal{L},\mathcal{D}_{test}}(f_\theta)$ is calculated as a representative of the expected risk $R_{\mathcal{L},\mathcal{D}}(f_\theta)$.

Given a model f_θ, its expected risk $R_{\mathcal{L},\mathcal{D}}(f_\theta)$ would be low when f_θ fits the data in an optimal way. Generally speaking, a machine model cannot fit the data perfectly mainly due to two reasons including overfitting and underfitting, as demonstrated in Fig 2.2. Overfitting and underfitting may exist simultaneously in some real-world applications.

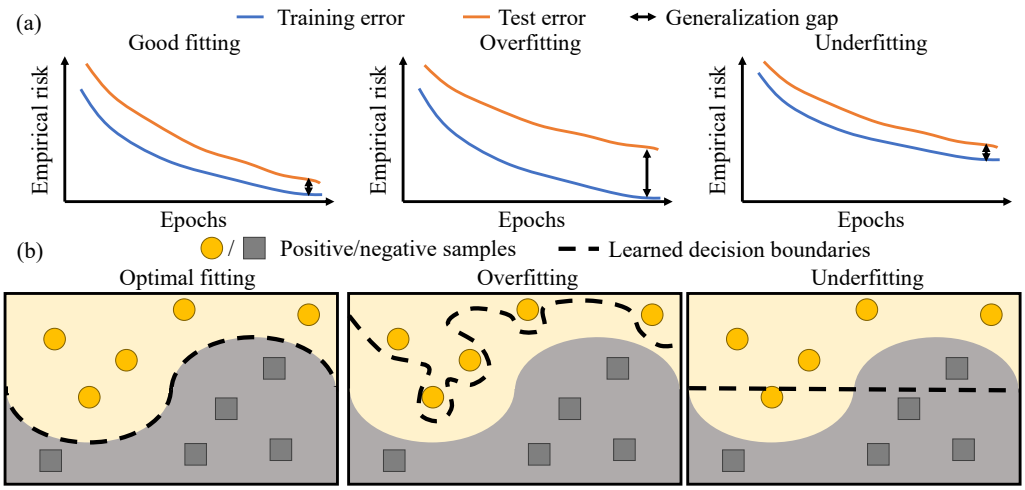


Figure 2.2: The schematic diagrams of good fitting, overfitting and underfitting for the classification problem. (a) The learning curves of different situations. In practice, the generalization gap is calculated as the discrepancy between the training and the test errors. The model test error cannot be decreased due to either overfitting or underfitting. (b) The feature embeddings and decision boundaries of different situations. When the model overfits the training datasets, it would build an over complex decision boundary very close to the training samples but cannot generalize well on unseen test data. On the contrary, the model would fail to correctly classify representative training samples with an over simplified decision boundary due to underfitting.

2.1.2.1 Overfitting

The generalization gap $\mathcal{G}$ of the model f_θ is defined as the discrepancy between the empirical risk of the test and training data:

$$\mathcal{G} := R_{\mathcal{L},\mathcal{D}_{test}}(f_\theta) - R_{\mathcal{L},\mathcal{D}_{train}}(f_\theta). \tag{2.6}$$

In the case of overfitting, the model can perform well on the training data but cannot adapt properly to the test data. As illustrated in middle part of Fig. 2.2(a), the test error of the overfitting model is high even though the training error is reasonably low.

Overfitting happens when a model has many parameters and the training data is under-represented. For example, modern DNNs would contain a large amount of network parameters, which can feasi-

bly memorize everything in the training data including noise [256]. In most cases of deep learning, the generalization errors cannot be decreased because the model overfits the training data. Specifically, the decision boundary of the DNNs would be affected by the biases introduced by noise or spurious patterns in the training data, as described in the middle diagram of Fig. 2.2(b).

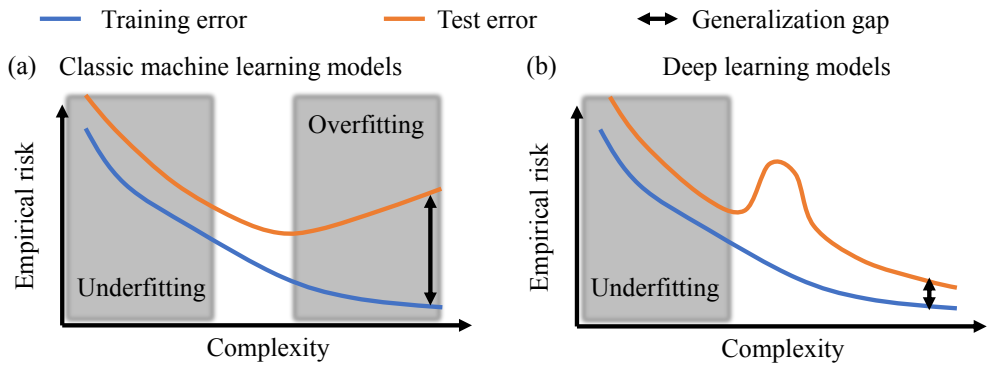


Figure 2.3: The generalization gaps when increasing model complexity of different machine learning models. (a) Classic machine learning models are prone to overfit the training datasets when containing too many parameters. (b) However, deep learning models are found to have the phenomenon of double descent in the generalization gaps when adding model complexity. In other words, deep learning models are unlikely to suffer from overfitting as a result of increasing model capacity.

2.1.2.2 Underfitting

When a model underfits the training data, it would fail to fit the data well, failing to reduce the empirical risk of the training data $R_{\mathcal{L},\mathcal{D}_{train}}(f_\theta)$. In this case, even though the model has relatively small generalization gap, it cannot perform well on the unseen test data. As shown in the right part of Fig. 2.2(a), the training and test errors remain high during the training process.

A neural network would underfit the training data when f_θ is simple and the training data distribution is difficult to model. In this case, the model cannot extract discriminative features to represent the training data, thus cannot build good decision boundaries to correctly classify the training samples. As illustrated in the right part of Fig 2.2(b), an underfitted model makes predictions only using weak features and fails to recognize patterns hidden behind the training data.

2.1.2.3 Generalization Capacity

The generalization capacity of DNNs is the model's ability to perform well on the previously unseen test data $\mathcal{D}_{test}$. Improving the generalization capability of neural networks is an important and fundamental problem of deep learning. It can lead to reliable model and help us achieve optimal results for unseen data with better understanding [184]. Identifying the factors that can affect

model generalization capacity can help us design the model for tasks and datasets with specific complexity.

Classic statistical learning theories such as Rademacher complexity [11] and the Vapnik–Chervonenkis (VC) dimension [234] analyze the generalization based on a low-capacity class of hypotheses. From the perspective of small class capacity, the generalization capacity is determined by many factors including the real data distribution $\mathcal{D}$, the training data distribution $\mathcal{D}_{train}$, the model f and the hypothesis space $\mathcal{F} \ni f$. For example, the generalization gap would become larger as the model complexity, which is the VC dimension of $\mathcal{F}$, increases. As a result, traditional learning theory believes that a machine learning model should be as simple as possible to have good generalization [11, 234], as shown in Fig 2.3(a). That is to say, the simplest model which has the equally lowest $R_{\mathcal{L},\mathcal{D}_{train}}(f_\theta)$ should have the lowest $R_{\mathcal{L},\mathcal{D}_{test}}(f_\theta)$.

The above mentioned statistical learning theories provide a theoretical bound and probability to analyze the generalization of neural networks. In contrast, recently many empirical studies on deep learning reveal that DNNs would not suffer from the generalization drop no matter how much parameters the model have [185, 195]. Their experiments demonstrate that the more parameters the model has, the better the performance is and the lower the generalization gap is, although there is not a good explanation for this phenomenon. Specifically, empirical results show that the generalization gap would become large as the model complexity increase in some regions, but eventually would be small as the model complexity is large enough [180]. We illustrate this double descent phenomenon in Fig 2.3(b). In this case, the generalization capacity of DNNs seems to be completely determined by the real data distribution $\mathcal{D}$, the training data distribution $\mathcal{D}_{train}$ and the model f [120, 256]. The model complexity would no longer affect the generalization gap.

Model generalization generally is influenced by the penalties of model complexity and the penalties of data complexity. As above mentioned, the impact of model complexity on DNNs remains unclear and difficult to be optimized. In contrast, the impact of dataset complexity is clear and intuitive: the more data we get, the noise would have less impact on the training data, thus the generalization gap would be smaller [224]. Therefore, in this thesis we aim to provide alternative algorithms to improve the generalization capability of neural networks by reducing the penalties of dataset complexity, which is related to regularization techniques such as data augmentation.

2.1.3 Meta-Learning

Meta-learning, or learning to learn, is a learning strategy that optimizes high-level objectives with hyper networks [209]. Instead of solving the specific tasks with pre-configured settings, meta-learning can modify the model architectures and hyper-parameters to fit the tasks well. We illustrate the idea of meta-learning in Fig. 2.4. Meta-learning optimizes the meta-knowledge ω such that the task objectives can be of benefit. The parameter ω can be designed to represent a wide range of options or matters related to the learning process. Specifically, ω can be explicitly set as hyper-parameters

such as learning rate [70, 140], the choices of optimizer [174], cost functions [144], data augmentation [52] and network architectures [28, 157]. Moreover, meta-learning can also implicitly learn to initialize the network parameter θ such as the model can produce generalizable features [73, 220].

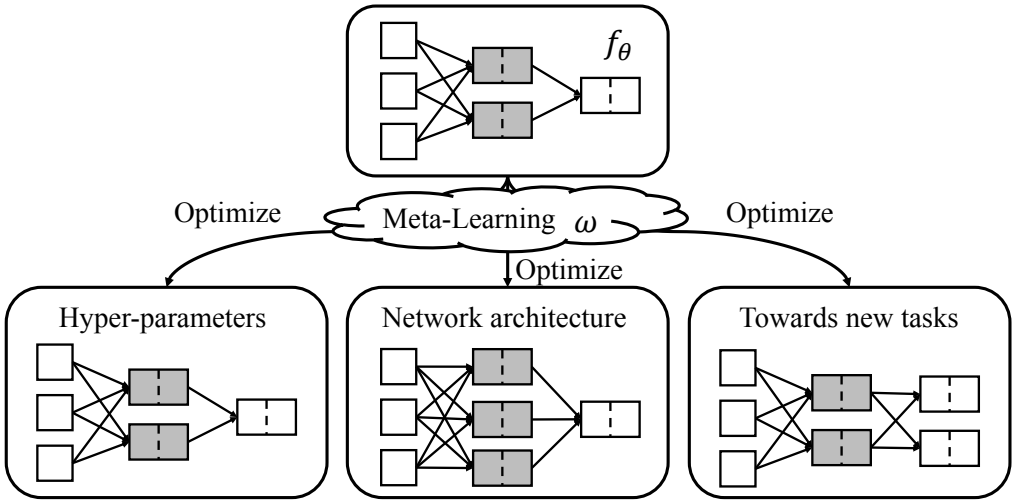


Figure 2.4: The illustration of meta-learning. Traditional machine learning algorithms solve specific tasks with pre-configured models. On the contrary, meta-learning learns the way to solve machine learning problems. In other words, meta-learning aims to optimize the algorithm itself for better generalization. Specifically, meta-learning can learn to optimize hyper-parameters, network architecture for given tasks. Furthermore, it can explicitly improve model generalization towards new tasks without fine tuning.

The meta-learning task is always taken as a bi-level optimization problem, where one optimization process is constrained by the other optimization [102]. Given a meta-training dataset $\mathcal{D}_{meta-train} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$ and a meta-test dataset $\mathcal{D}_{meta-test} = \{(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{y}}_i)\}_{i=1}^{M}$, the objectives of meta-learning can be formulated as:

$$\min_{\omega} \frac{1}{M} \sum_{i=1}^{M} \mathcal{L}_{val}(f_{\theta^*}(\tilde{\boldsymbol{x}}_i), \omega, \tilde{\boldsymbol{y}}_i) \tag{2.7}$$

$$s.t. \quad \theta^* = \underset{\theta}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{train}(f_\theta(\boldsymbol{x}_i), \omega, \boldsymbol{y}_i), \tag{2.8}$$

where $\mathcal{L}_{val}$ is the cost function related to high-level objectives and $\mathcal{L}_{train}$ is the cost function used to update the model parameter θ. Note that for some cases, $\mathcal{D}_{meta-test}$ can be the same as $\mathcal{D}_{meta-train}$ [9].

The meta-learning process is always divided into two procedures including the inner and outer loop.

The inner loop shown in Eq. 2.8 is conditioned on ω to achieve a local optimal θ^*. The outer loop described in Eq. 2.7 learns ω such that θ^* performs well on $\mathcal{L}_{val}$ after training with $\mathcal{L}_{train}$.

The optimization process is typically done in the way of epoch training. During each epoch, $\mathcal{D}_{meta-train}$ and $\mathcal{D}_{meta-test}$ are sampled separately from the data distribution. The outer loop optimization would be operated after the inner optimization function is satisfied. The outer optimization could contain second order optimization and can be solved based on gradient [73], reinforcement learning [52] or evolution [209, 223]. Gradient based optimization methods calculate $\frac{\partial \mathcal{L}_{val}(f_{\theta^*}(\tilde{\boldsymbol{x}}_i),\omega,\tilde{\boldsymbol{y}}_i)}{\partial \omega}$ based on the chain rule of differentiation. Optimizing the outer loop based on gradient makes the process efficient but require the meta-objective to be differentiable. On the contrary, the optimization methods based on reinforcement learning and evolution can deal with meta-objective that is not differentiable but require more time and might need additional efforts to make sure the optimization process can converge. Therefore, reinforcement learning and evolution based optimization are more commonly adopted to reinforcement learning applications which always contain multiple non-differentiable steps [13, 103].

2.2 Regularization in Deep Learning

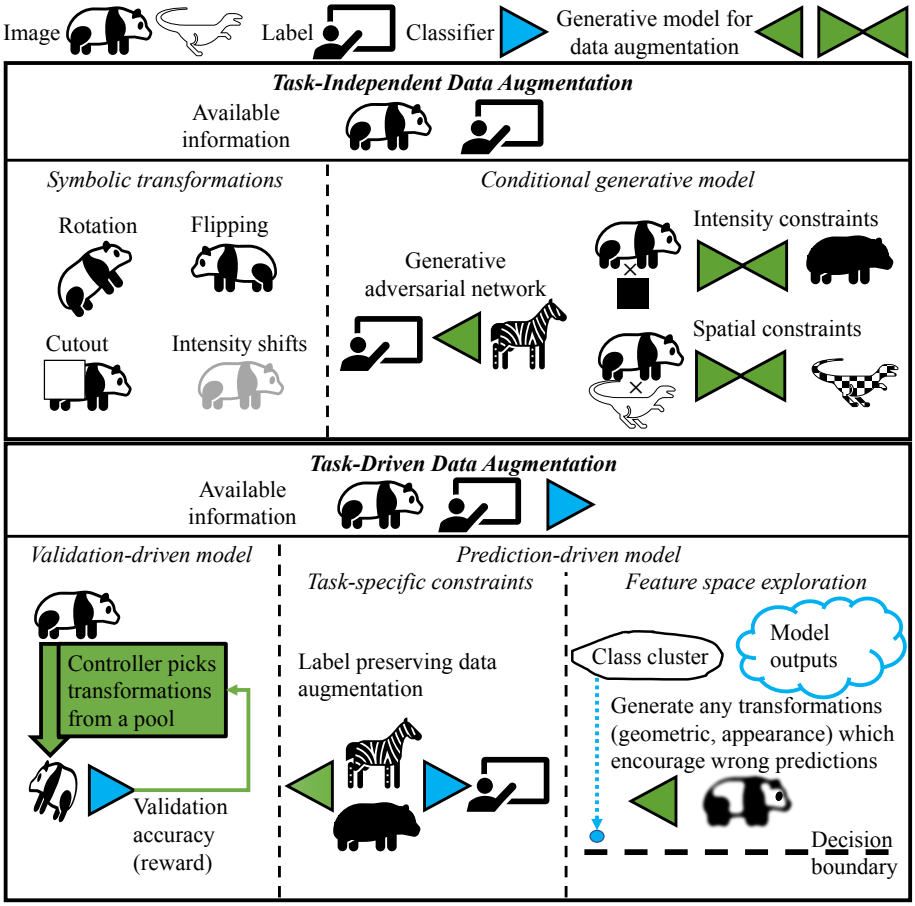


Figure 2.5: An overview of different data augmentation methods for image processing. In the view of whether classifiers are involved in generating synthesized samples, data augmentation approaches can be categorized into task-independent data augmentation and task-driven data augmentation. For task-independent data augmentation, it can be further divided into symbolic transformations and conditional generative model according to whether a generative model is utilized to produce the synthesized samples. For task-driven data augmentation, it can be further divided into validation-driven model and prediction-driven model based on whether the accuracy of a hold-out dataset is taken into account.

In order to improve the generalization capability of neural network from the data perspective, many regularization techniques have been proposed. We will discuss data augmentation, which is one of the major concerns of our study, in detail in the following section and introduce other data based regularization techniques in section 2.2.3.

Data augmentation, which aims to enlarge the dataset by adding synthesized training data using varied transformation $\mathcal{T}(\cdot)$, is a common practice in machine learning applications and has been

proven to steadily improve the generalization [233].

As summarized in Fig. 2.5, according to whether the classification model is involved to produce the synthesized samples, recent data augmentation methods for deep learning models can be categorized into two groups: 1) Task-independent data augmentation, of which augmented samples are generated by only using the images and task independent. 2) Task-driven augmentation, of which the augmented samples are generated by taking use of the gradients from the training of specific tasks.

2.2.1 Task-Independent Data Augmentation

Data augmentation policies are often pre-designed based on heuristics and kept the same during the model training. According to the formulation of the transformation $\mathcal{T}(\cdot)$, task-independent data augmentation can be further categorized into symbolic transformations based or conditional generative model based data augmentation.

2.2.1.1 Symbolic Transformations

Hand-designed transformation is a well-studied data augmentation method and can be applied to computer vision tasks along with neural networks from the very beginning. The transformations are designed to mimic possible variants of unseen data based on human knowledge. It has been proven to be widely effective but not much changes are made among different tasks for following-up studies. Intuitively, the noise can be added to the input images to make the model stronger and more robust to perturbations [20]. Conventionally, the training samples are augmented based on transformations including rotations, flipping, translation, projection, inverting, shearing, contrast-changing, color-changing, brightness-changing, sharpness-changing, solarization, posterization, histogram matching, intensity shifts and nonlinear deformations [49, 127]. These transformations would simulate the test conditions under which the models would be deployed.

Recently, some simple but effective data augmentation strategies are proposed, such as cutout and super-pixel. The former method randomly pads the original images with gray patches, while the latter one enrichs the training dataset with samples using superpixel based pre-processing. They show evident improvements for image classification [60] and gland segmentation [262]. While simple, those heuristic approaches are widely adopted to improve generalization for varied applications.

The above mentioned data augmentation models adopt the same set of transformation for different classes. Except for those data augmentation stategies which would apply the same transformations for different samples from different categories, some methods generate additional samples based on the relationship between samples. In order to take advantage of inner-class relationship, synthesized samples can be generated by the co-registration of training samples from the same class [90]. Specifically, they align image pairs from the same class, with an assumption that the la-

tent spatial transformation between them s belongs to a large class of diffeomorphisms They show improvements on the MNIST datasets but the idea can be naturally extended to medical imaging. Taking consideration of inter-class relationship, mixup randomly combines two samples including images and labels in a linear way and is reported to show significant improvements for both image classification [261] and image segmentation [66]. Although mixup seems to be a counterintuitive transformation, but it works as it could locally linearizes the landscape of the loss.

In order to add more variations and realistic transformations to the training data, a group of studies try to include more knowledge which is distilled from the unlabeled images. Some data augmentation methods can be extended to the setting of semi-supervised learning with minor modifications. In order to enlarge datasets based on fair symbolic transformations, synthesized images can be generated by combining the content of labeled data and the the intensity and spatial appearances of unlabeled data [263]. The intensity and spatial transformations are learned in pairwise comparison by two separate models. The ground truth labels are transformed in the same way. The authors [263] validate their method on the task of one-shot brain segmentation and show that the augmented samples can improve the brain segmentation performance.

2.2.1.2 Conditional Generative Model

Generative models, such as generative adversarial network (GAN), variational autoencoder (VAE) and diffusion models, aim to model the data generating process and have shown strong ability to generate synthetic but realistic and diverse samples. Some methods extend the training datasets using samples produced by generative models. In order to synthesize samples for a certain class, GAN produces realistic samples such that it can fool a discriminator model. Specifically, GAN is trained to maximize the cost function of the discriminator model and minimize the cost funciton of the generative model to figure out the generated samples from real samples. The generated samples can boost the network performances on many image classification datasets [3].

Similar ideas are applied to the classification of liver lesion classifications [75]. Different liver lesion regions of interest (ROIs) are synthesized with the same size from the input noise. In order to generate more stable and realistic samples, an image-to-image generative model which is trained to map non-nodule regions to nodule regions is adopted to improve the segmentation of lung [114]. Similar image translation approach is adopted by [85] to improve the performance of bone lesion classification. Assuming that not all the generative samples are usefully, poorly generated samples are filtered out based on the reactions of the normally trained classifier, showing further performance improvements.

Conditional generative models can be used to generate samples by building bridges between labeled and unlabeled images. The utilization of unlabeled data for generative models is also straight forward. [34] adopt a conditional GAN to generate realistic training samples based on intensity and spatial transformations. The discriminator is trained to differentiate the generated samples from

both the labeled images and unlabeled images. They report that the model can largely benefit from the augmented samples for cardiac segmentation. Assuming that the symbolic transformations rarely map between classes, [200] proposes to learn a series of realistic transformation functions just using unlabeled data based on GAN. The transformations which can make the images look the most realistic are applied to the labeled data. They report that the realistic transformations can outperform totally random symbolic augmentations.

2.2.2 Task-Driven Data Augmentation

The above mentioned data augmentation methods generate new training samples without considering specific tasks. That is to say, the data augmentation strategies can be the same for different tasks with the same dataset. However, a more effective way is to produce specific and effective samples for the given tasks. This data augmentation process can be integrated along with model training.

Recently, many methods have made attempts to generate task-driven data augmentation by taking consideration of the network output of the augmented samples. The schemes of these task-driven data augmentation algorithms can be roughly categorized into two groups, according to whether the accuracy of a held-out dataset is considered in the optimization process. Validation-driven models utilize another model to adjust the data augmentation strategies based on the model performance on the held-out dataset. On the contrary, another group of methods aims to learn transformations for data augmentation based on the prediction of training data, with the criterion of label preserving or controllable hardness.

2.2.2.1 Validation-Driven Model

In practice, data augmentation policies are usually adjusted based on the model performance on the validation dataset. Typically, the data augmentation parameters could be obtained by grid search based on the validation performance [38]. Motivated by this process, the empirical risk on a held-out dataset is explicitly formulated as the cost functions to optimize the data augmentation. Specifically, reinforcement learning algorithms are utilized to find the transformations which can make the model generalize well. These methods work well with symbolic transformations as they do not require the transformations to be differentiable. In addition, the optimization of discrete variables is easier as it simplifies searching space.

[52] proposes AutoAugment to learn data augmentation strategies based on reinforcement learning and created new SOTA results on several authoritative image classification datasets including CIFAR, SVHN and ImageNet. Although their results are quite impressive, their idea is actually very simple: to use a searching algorithm to find the best augmentation policies which can make the model perform better on a separate validation data. Interestingly, although they included 14 symbolic transformations in their study, the finally results are achieved just by using some classic

transformations such as rotation and histogram matching instead of more complicated transformations such as cutout or mixup. Follow-up works reduce the optimization time of AutoAugment by density matching [89, 153] or meta-learning [145].

2.2.2.2 Task-Specific Constrains

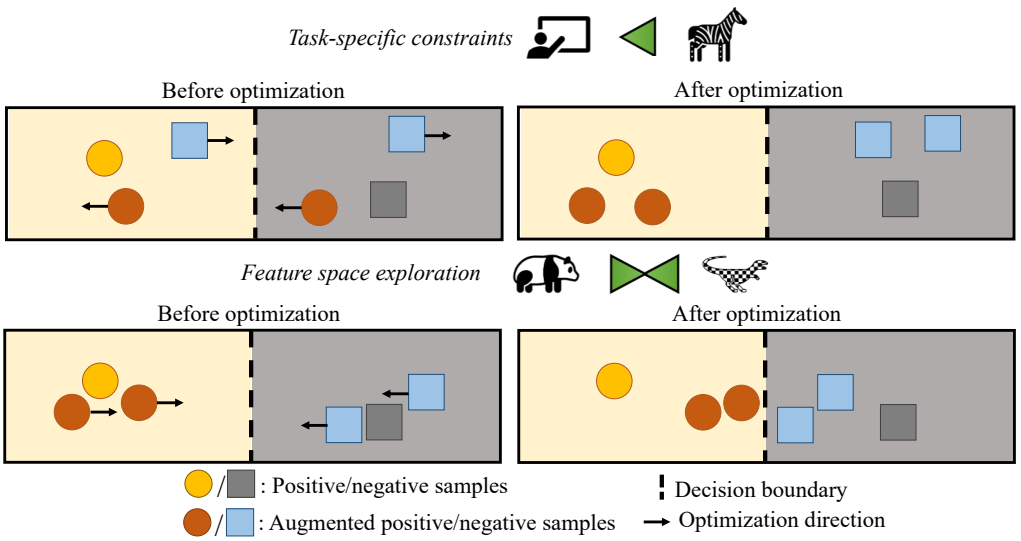


Figure 2.6: The comparison of different prediction-driven model based data augmentation methods. Task-specific constraints encourage the generated samples to be well-classified and representative for the given tasks. On the contrary, feature space exploration aims to generate difficult samples to build robust decision boundaries.

Currently there are totally two different ways to utilize the supervision from the classifier based on different assumptions, as demonstrated in Fig. 2.6. These methods are usually built with generative models because the generator can be adjusted by optmizing the loss and following the gradient. In this way, the data augmentation process can be feasibly connected with the training of the desired classifier. The first group, of which samples are usually generated from noises, assumes that the generated samples are quite random and need to be regularized. Therefore, they encourage the generative model to produce the augmented samples which can be correctly classified. The second group, of which samples usually generated from prior information, assumes that the generated samples should stay near the original images. Therefore, they largely explore the feature space by adversarial loss or Kullback–Leibler (KL) divergence to encourage the generative model to produce samples which are closer to the decision boundary and hard to classify.

When the generated samples can be produced just from noises, the samples have little class-conditional properties and lack fidelity. Therefore, many methods try to involve the supervision of classifier networks into the training process to both generate class-conditional images and improve the performance of the classifier. Developed upon the backbone of conditional GAN, [227]

learned the data distribution of the training dataset using the supervision of classifier. Then they treats the synthesized data as missing variables and generates the samples based on the learned distribution. A similar triple-network architecture is also proposed in [46] but in a semi-supervised way. They both show evident improvements on image classification.

Task-specific constrains can also help reduce the noisy samples from the generated images. In this case, the classifier is separately trained and just serves as an auxiliary task. Although the generative model is built based on image-to-image architecture which can generate stable images, [34] encourages the generative model to produce well-classified images by using the supervision of augmented samples. Similarly, in order the guarantee that the generated samples have enough lesion-like properties, [85] chooses the generated samples which have high responses when predicted as the lesion class.

2.2.2.3 Feature Space Exploration

In another aspect, some methods generate synthesized samples to explore the feature space. The synthesized samples are generated by optimizing with the inverted gradient from the classifier in the input space. Specifically, they try to improve model performance by training with hard samples, which have high classification loss when being processed by the network. Most of the studies are developed based on the idea of adversarial training [79], of which adversarial samples are generated by adding small but intentionally worst-case perturbations to the input images. The adversarial training samples are generated based on reverse gradient [177]. Adversarial training encourages the neural network to behave constantly within the local neighborhood. Specifically, a function g parameterized by ω is utilized to transform the input image $\boldsymbol{x}_i$. The ω_{adv} is optimized to maximum the cost function of the generated samples:

$$\omega_{adv} = \underset{\omega;\|\omega-\omega_{adv}\|<\epsilon}{\operatorname{argmax}} \mathcal{L}(f_\theta(g_\omega(\boldsymbol{x}_i)), \boldsymbol{y}_i), \tag{2.9}$$

where ϵ is a small vector that restricts the variance of the generated samples. In practice, ω_{adv} is obtained with one or a few steps of gradient ascent. In order to make sure the generated samples are from the same distribution with the original data, g_ω is formulated as simple functions such as addition or affine transformations [41]. [190] produces adversarial samples which near the decision boundary based on geometric transformation. They show improved accuracy and robustness in the task of skin lesion and breast tumor classifications. [215] proposes to generate adversarial samples in the feature space. The adversarial features which can make the classifier uncertain about the prediction are generated. In addition, they utilize an attention mask based on entropy to encourage the network to focus on the regions which are the most relative with the classification task. They show SOTA results on few-shot classification tasks.

[193] learns to select useful transformations including scaling, rotating and occluding in the training

processing of the classification model. In each training epoch, they first input random transformed images into the network and then find the most hard augmentation samples to train the network. The hard examples are defined as the augmented samples which can enlarge the KL divergence of the network output of the augmented and original samples. They show that their augmentation training methods can improve the accuracy of human pose estimations.

Adversarial training can provide the direction towards the decision boundary, however, it cannot exactly decide the location of the adversarial samples in the feature space. On the contrary, some studies generated samples with certain feature location by making changes directly in the feature space. [264] synthesize samples in the features space based on the relationship of a pair of samples from different classes. First, they generate feature samples based on the linear interpolation of a negative pair. Then, they build another generator to map the features back to the previous feature space with label-preserving constraints. Finally, the augmented features are utilized to train the network. They show that this method can exploit more information buried in the training samples and provided competitive results in some image classification tasks.

2.2.3 Other Regularization Approaches

Following the idea of choosing the appropriately simple architecture for neural networks, many practical regularization methods such as early stopping [20], residual connections [93], weight decay [128], parameter constrains [221], dropout [222] and spectrum normalization [251] have been proposed from the model perspective.

In addition to data augmentation, many regularization techniques have been explored to improve the generalization capability of neural networks from the data perspective.

Besides the input space, adding noise can be also added to other places of the models to improve robustness. Noise can be added to the weights to enhance the model ability to handle uncertainty [183]. In addition, noise can be included in the labels and therefore generate soft labels. Label smoothing can prevent the pursuit of the fitting with hard probabilities and extreme cases. For example, in image classification the one-hot encoded labels can be degraded by a factor instead of being 1 to improve the generalization capability [183].

Model ensemble is a widely adopted technique to improve model generalization, which averages the prediction results from multiple models. As being initialized with random weights, neural networks always obtain different optimal parameters even with the same training dataset [129]. Averaging model predictions is empirically powerful and reliable because it can remove inconsistent errors [24].

Embedding more knowledge by adding more information in the input or output space is another way to improve generalization. More information can help the model gain stronger statistical strength [12]. For example, multi-model images can be fed together into the neural network to help

the network understand the image better and avoid local optimum [173]. More recently, the joint modeling of images and languages [199] has shown promising results in image generation.

Multi-task learning requires a model to learn different tasks at the same time to make the model regularize well by fitting different tasks with shared model parameters [32]. In multi-task learning, neural networks are driven to build more generic features, if the tasks are chosen properly [254].

When additional unlabeled data are available, semi-supervised learning can improve the generalization by encoding more prior image information [36]. Semi-supervised learning encourages the neural network to learn better representation by prior constrains such as entropy minimization [17] and consistency regularization [131, 245].

2.3 Image Segmentation

Image segmentation is a computer vision task by which the boundaries of ROIs in images are delineated through machine learning algorithms. Image segmentation can partition an image into several regions with specific semantic meanings, making the image easier to analyze for downstream tasks such as autonomous driving [51], disease diagnosis [130] and computer assisted surgery [69]. Image segmentation can also be an objective itself such as visualization.

This study focuses on the problem of medical image segmentation considering its high clinical relevance and challenging nature. Medical image segmentation aims to identify the anatomical structures or other ROIs in medical imaging such as magnetic resonance imaging (MRI) or computerised tomography (CT). We show an example of brain tumor segmentation in MRI from varied registered sequences using data from brain tumor segmentation (BraTS) challenge in Fig. 2.7. The segmentation of brain tumor can help evaluate the progress of the disease and the success of a chosen treatment [101]. However, the segmentation process is always challenging and time-consuming as the ROI appearance is heterogeneous while the three-dimensional (3D) imaging volume is huge. When the imaging data are collected with thick slices, manual delineation is can be rather inaccurate. Automating the procedure of medical image segmentation can save doctors a great deal of clinical time.

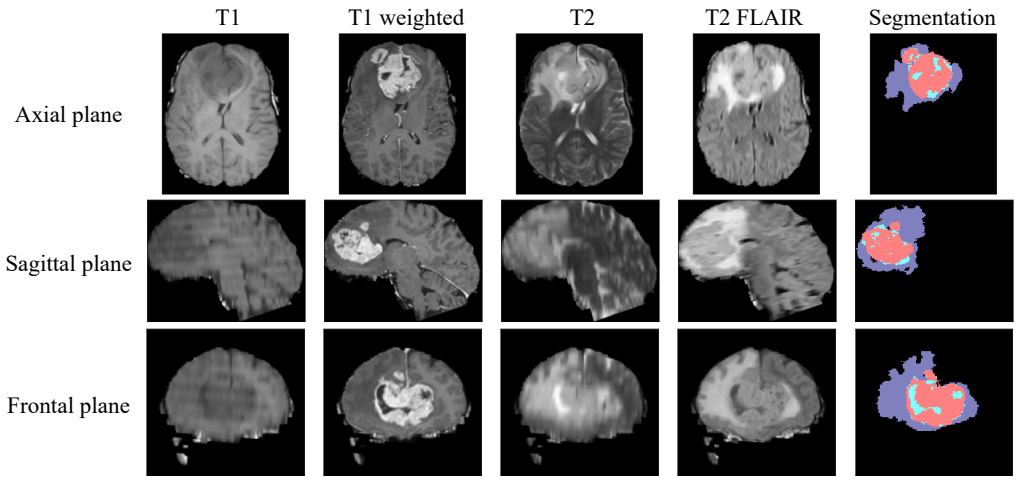


Figure 2.7: An example of medical image segmentation. We visualize the task of brain tumor segmentation using data from BraTS challenge. This task requires to segment the three classes of brain tumors based on four 3D MRI sequences. We show magnetic resonance (MR) images together with their corresponding segmentation in three planes. Different colors represent different classes of brain tumors.

2.3.1 Medical Image Segmentation

Medical image segmentation, as one of the main medical image computing tasks, provides critical clinical information and serves as a prerequisite for subsequent image-base diagnosis process [206]. It has been an important research focus for decades. Many methods have been proposed to make this process semi-automatic or fully automatic, with an aim for high accuracy and efficiency.

Early attempts for solving medical image segmentation are mostly based on deformable models [169] such as active contour [246] which aim to match predefined deformable models based on energy minimization. By being a deformable models, active contour models are robust and can automatically adapt to images with different conditions. However, they are sensitive to initial states and local minima states. Level-set method [138] was thereafter proposed to reformulate the curve evolution function with level set equations, facilitating the models to follow shape that change topology without having to paramterize these objects. Level-set method is computational expensive as it solves the curve evolution problem with a higher dimension function.

Statistical models [194] including Markov random field (MRF) [241] are also utilized to delineate the object outlines by modeling the spatial interactions between nearby pixels. Graph cut [22] partitions an image into sub-regions by maximizing the flow in the graph, which minimizes energy in a form of MRF. These methods are known to straggle to fit a small contour as they are optimized to find a minimum cut.

The anatomical structures are consistent between different subjects. Therefore, registration techniques [105] are widely adapted to segmentation by registering the target images to different atlases [2]. The segmentation results are obtained by wrapping the atas segmentation maps with the reversed transformation fields. These registration based segmentation methods are robust and effective for the segmentation of objects with consistent structures such as brain and cardiac, however cannot segment objects of which appearances vary case to case, such as tumor and lesion.

The segmentation task can be seen as dense prediction on pixels. In this way, discriminative models such as random forests [25] and sparse representation [242] can be utilized to make predictions on features with sliding windows. The effectiveness of these classical feature extraction techniques highly relies on the quality of the crafted features. However, the optimal features, might differ from case to case. More recently, the strong recognition ability of DNN in image classification has been transferred to image segmentation. Neural networks learn the features and the classifier together during the training process, therefore the features are always tailored to the given tasks. DNN has been successfully adapted to varied medical image segmentation [178, 204], showing SOTA segmentation accuracy with high speed. Therefore, this thesis would focus on segmentation with neural networks.

2.3.2 Segmentation with Neural Networks

2.3.2.1 Convolutional Neural Networks

Neural networks such as MLPs are not efficient for computer vision applications where data are always high-dimensional. We illustrate an example of calculating a hidden layer with an image in Fig. 2.8(a).

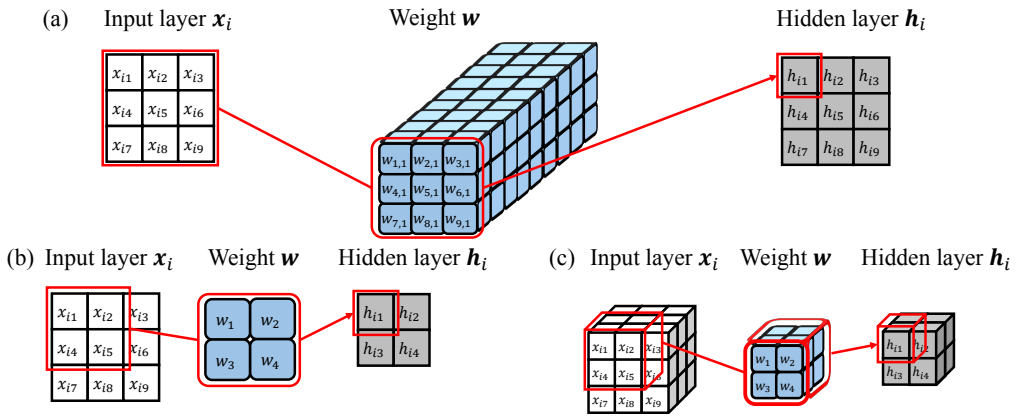


Figure 2.8: The illustration of a convolutional layer in CNN processing two-dimensional (2D) inputs. (a) Calculating the hidden layer $\mathbf{h}_i$ with 2D input layer $\mathbf{x}_i$ using MLP. As all input elements are connected to calculate the output elements, the parameter $\mathbf{w}$ would be huge and the calculation would be expensive when $\mathbf{x}_i$ becomes larger. (b) In contrast, calculating the hidden layer $\mathbf{h}_i$ using convolutional layer is computationally more efficient. As $\mathbf{h}_i$ is calculated with sliding window, the parameter $\mathbf{w}$ can stay small regardless of input size. (c) The convolutional layer would be relatively more efficient when compared with MLP for 3D images. Note that here we omit all biases $\boldsymbol{b}$ for simplicity.

In image processing, the responses of neurons when processing an 2D features are named as features maps. Specifically, the feature map of the hidden layer $\boldsymbol{h}_i$ is calculated as:

$$\boldsymbol{h}_i = \begin{bmatrix} h_{i1} \\ h_{i2} \\ h_{i3} \\ h_{i4} \\ h_{i5} \\ h_{i6} \\ h_{i7} \\ h_{i8} \\ h_{i9} \end{bmatrix} = \begin{bmatrix} w_{1,1} & w_{2,1} & w_{3,1} & w_{4,1} & w_{5,1} & w_{6,1} & w_{7,1} & w_{8,1} & w_{9,1} \\ w_{1,2} & w_{2,2} & w_{3,2} & w_{4,2} & w_{5,2} & w_{6,2} & w_{7,2} & w_{8,2} & w_{9,2} \\ w_{1,3} & w_{2,3} & w_{3,3} & w_{4,3} & w_{5,3} & w_{6,3} & w_{7,3} & w_{8,3} & w_{9,3} \\ w_{1,4} & w_{2,4} & w_{3,4} & w_{4,4} & w_{5,4} & w_{6,4} & w_{7,4} & w_{8,4} & w_{9,4} \\ w_{1,5} & w_{2,5} & w_{3,5} & w_{4,5} & w_{5,5} & w_{6,5} & w_{7,5} & w_{8,5} & w_{9,5} \\ w_{1,6} & w_{2,6} & w_{3,6} & w_{4,6} & w_{5,6} & w_{6,6} & w_{7,6} & w_{8,6} & w_{9,6} \\ w_{1,7} & w_{2,7} & w_{3,7} & w_{4,7} & w_{5,7} & w_{6,7} & w_{7,7} & w_{8,7} & w_{9,7} \\ w_{1,8} & w_{2,8} & w_{3,8} & w_{4,8} & w_{5,8} & w_{6,8} & w_{7,8} & w_{8,8} & w_{9,8} \\ w_{1,9} & w_{2,9} & w_{3,9} & w_{4,9} & w_{5,9} & w_{6,9} & w_{7,9} & w_{8,9} & w_{9,9} \end{bmatrix} \begin{bmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \\ x_{i4} \\ x_{i5} \\ x_{i6} \\ x_{i7} \\ x_{i8} \\ x_{i9} \end{bmatrix}. \quad (2.10)$$

As MLPs calculate the output at each pixels by using all the input pixels, the networks parameters would increase in a square number as the size of a 2D input image increases. Moreover, the computational complexity when processing an 3D image would grow cubically with large image size. Fully connected weights contain much redundant details, which are not necessary for recognition as significant image features always exist only in specific regions of the images. In addition, the large amount of network parameters make it inefficient for the optimization and inference of neural networks. Therefore, MLPs, which are multiple fully connected layers, are always only utilized on the top of extracted features as a classification layer. Instead, the features are always extracted by a convolutional layer, which is the core component of CNNs.

Most neural networks based segmentation methods are developed based on CNNs, because of its high computational efficiency and strong capability of recognition, especially when working with high-resolution images. Different from an MLP, a convolutional layer calculates the feature maps using convolutional kernels with fixed kernel size. During the forward propagation, the feature maps are obtained with the dot product between the input and the filter kernels. As illustrated in Fig. 2.8(b) and Fig. 2.8(c), the whole feature maps are generated by sliding the convolutional kernel along the input features with shared weights. Therefore, feature maps can be calculated using the same convolutional kernels for images with arbitrary image size. Given the same input image $\boldsymbol{x}_i$, the feature map $\boldsymbol{h}_i$ is calculated with:

$$\boldsymbol{h}_i = \begin{bmatrix} h_{i1} \\ h_{i2} \\ h_{i3} \\ h_{i4} \end{bmatrix} = \begin{bmatrix} w_1 & w_2 & 0 & w_3 & w_4 & 0 & 0 & 0 & 0 \\ 0 & w_1 & w_2 & 0 & w_3 & w_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & w_1 & w_2 & 0 & w_3 & w_4 & 0 \\ 0 & 0 & 0 & 0 & w_1 & w_2 & 0 & w_3 & w_4 \end{bmatrix} \begin{bmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \\ x_{i4} \\ x_{i5} \\ x_{i6} \\ x_{i7} \\ x_{i8} \\ x_{i9} \end{bmatrix}. \tag{2.11}$$

We can find that the learnable parameters in Eq. 2.11 are much less than the ones in Eq. 2.10, making CNNs less likely to overfit the training data. Padding could be utilized to keep the dimensions of the input and output layer the same, which refers to the addition of zero valued pixels to the boundaries of the input feature map. For example, we can add 1-pixel pad to the input layer $\boldsymbol{x}_i$ in Fig. 2.8(b) to make it 4×4, yielding the output layer $\boldsymbol{h}_i$ with size 3×3. Note that an MLP which fully connects features from different dimensions can be seen as a convolutional layers with 1×1 kernels.

In order to reduce the computation costs, the size of feature maps can be intentionally reduced by using pooling layers. Alternatively, convolving with stride larger than 1, which is the step length

when the convolutional kernels are applied to the features maps, can also reduce the size of features maps. Pooling layers do not contain any parameters that should be optimized. In contrast, pooling layers only downsample the feature maps with operations such as max or average. Similarly, the feature maps could be upsampled with unpooling layers where the features are interpolated to higher resolution.

The convolutional layer introduces some favourable inductive biases, which refer to the characteristics brought by the architecture design, for image recognition. Locality and translation equivariance are the two main inductive biases of CNNs. Locality means that the CNNs make the prediction of a pixel based on the features of its near neighbors. This characteristic can enhance the spatial consistency of the segmentation results. For example, CNNs are capable of identifying the pixels in tumor regions sharing similar appearance. As the feature maps are the filter responses of local features, a convolutional layer would produce the the same outputs with local features regardless of its spatial positions. This property refers to translation equivariance. This property reduces the risks of overfitting for CNNs as the object recognition is no more related to the object positions. For example, once the tumor regions are learned with CNNs, it can recognize tumor with varied locations and different context, even if the images are shifted or cropped.

2.3.2.2 Network Architecture

The research focus of visual recognition has shifted from feature engineering to architecture designing for better accuracy, efficiency and scalability [163]. A common limitation of CNNs is the lack of the ability to model long range spatial correspondence. The receptive field of CNNs refers to the area in input image which contribute to feature calculation in the last layer. A convolutional layer has small receptive field due to the small size of the convolutional kernels. As a result, the feature maps are calculated based on local patterns and cannot utilize the global information.

Increasing the receptive field of CNNs is one of the motivations for the design of network architectures. For example, the receptive field can be enlarged by increasing network depth [218], choosing dilated convlutional kernels [44] or utilizing attention mechanism [62]. DeepMedic [117] and 3D U-Net [204] are two popular network architecture designed specifically for medical image segmentation in 3D volume such as MRI or CT with large receptive fields.

We illustrate the architecture of DeepMedic in Fig. 2.9(a). DeepMedic gains large receptive field by utilizing multiple pathways to process the input images with different scaling. Specifically, the context is downsampled in three or five times and fed into separate network pathways. The information is then aggregated before the fully connected layers. DeepMedic can model the local pattern well as it contains multiple convolutional layers with high-resolution input images. However, the inference of DeepMedic is computationally inefficient as it processes images with small patches. In addition, the context knowledge might not be fully utilized as the downsampling process would cause the loss of information.

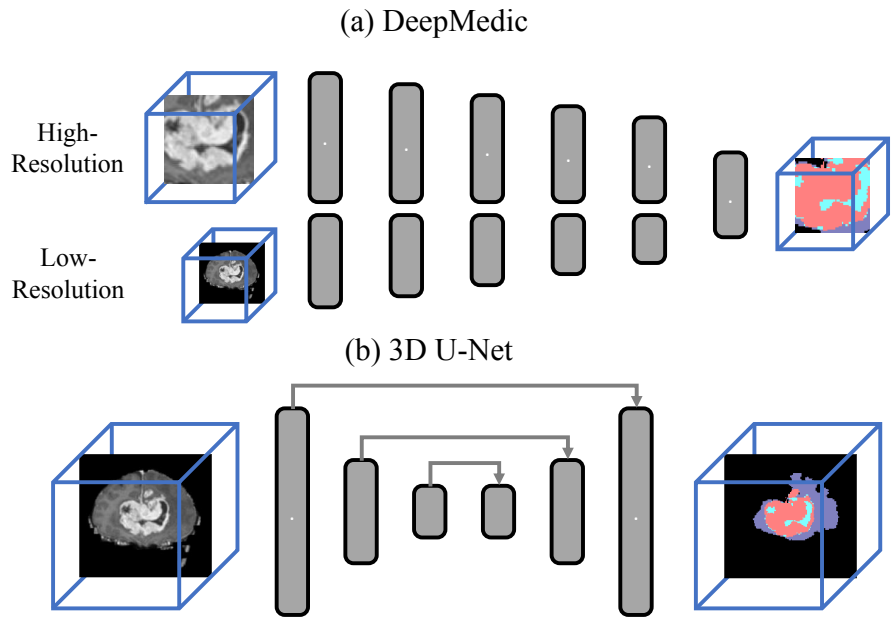


Figure 2.9: Two segmentation network architectures with successive convolutional layers. (a) DeepMedic enlarges the receptive field by taking use of low-resolution paths processing the large context. (b) 3D U-Net adopts a encoder-decoder architecture. The encoder consists of several convolutional and pooling layers, where the resolutions of features maps are decreased because of stride larger than 1. The decoder combines the upsampled features and intermediate high-resolution features of the encoder.

Another popular network architecture is U-Net, which is the most widespread segmentation CNN. We illustrate the architecture of U-Net in Fig. 2.9(b). U-Net utilizes an encoder to extract context information with a few downsampling steps. Its encoder-decoder architecture enables U-Net to extract features in an efficient way as the computation with the downsampled feature maps is less expensive. A key component of U-Net is the skip connection between the encoder and decoder, making it easier to optimize. U-Net is one of the most successful network architectures for medical image segmentation, still showing competitive segmentation performance after it was proposed for many years [106]. Due to its simplicity and effectiveness, many variants of U-Net including residual connection [271], dense connection [142], attention module [186] and transformer module [43] have been proposed afterwards.

Deep learning models are always designed for general purpose and can be applied to varied image segmentation tasks with minor modifications [106]. This study will utilize either or both architectures for the experiments.

2.4 Data Limitations in Medical Image Segmentation

Medical images usually have very imbalanced distributions for lesion segmentation and disease classification. In addition, when compared with natural images, medical images have different patterns and appearances. For examples, because medical imaging is complicated to be collected, it is more difficult to build large medical imaging dataset. Medical images sometimes have similar structures and features, therefore it is harder to distinguish one subtype from others.

These certain characteristics of medical imaging present obstacles to the optimization of neural networks. The learned neural networks could be biased because of noises but ignore the intrinsic patterns of medical imaging and lack generalization [239]. Thus, it is important to develop dedicated algorithms and build datasets with strong representative to improve generalization capability for medical image segmentation.

2.4.1 Class Imbalance

Class imbalance, which refers to the imbalanced classes of samples from different categories, is a common issue in image classification and image segmentation. We show some examples of medical image segmentation in Fig. 2.10. As the ROIs such as tumors and small abdominal organs are always rare, the foreground samples only occupy a small part of the 3D volume.

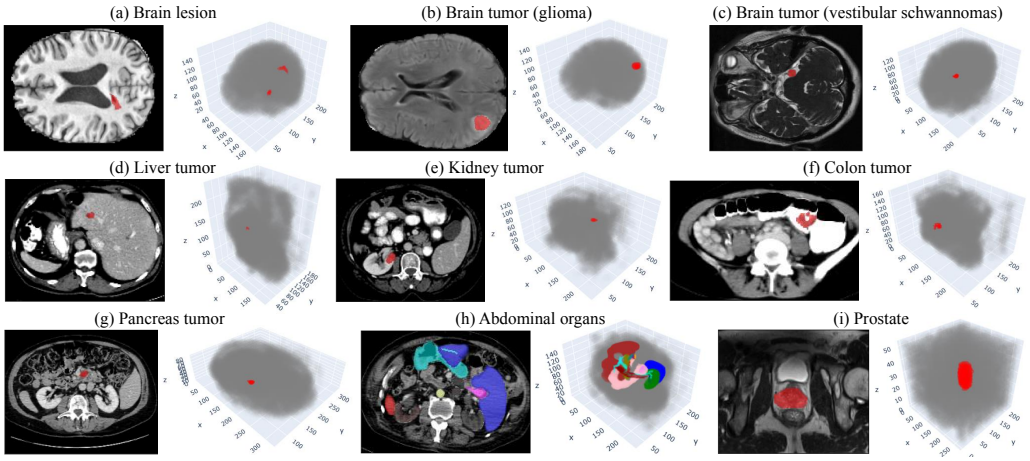


Figure 2.10: The illustration of class imbalance in medical image segmentation. We visualize both 2D slice and 3D scatter plots for all the datasets we study in this thesis. The ROIs, marked with red color for binary segmentation or varied colors for multi-class segmentation, are relatively small when compared with background samples.

The success of CNNs is strongly linked with the availability of large scale, representative datasets. However, in many real-world applications such as medical image segmentation, the availability of

large, annotated datasets is still limited. But even when there is a sufficient number of images available, the fundamental problem of class imbalance remains where ROIs (i.e. foreground classes) are heavily under-represented in the training data [26, 155].

Similar to [55], the class imbalance ratio of one image can be defined as the ratio between the number of pixels of the background class (which is commonly the most frequent class) and the number of pixels of different object classes. The class imbalance ratio of a whole dataset would then be reported as the average class imbalance ratio of all images in the set. Class imbalance ratios of 100:1 or higher are not uncommon in applications such as lesion segmentation, as shown in Table 2.1.

Table 2.1: The statistics and class imbalance ratios of the four datasets used in this study. Class imbalance ratio is defined as the average ratio between the number of the background (BG) pixels and the foreground (FG) pixels over all images. We obtain this statistic information based on the original images without any data augmentation.

Dataset	Total FG pixels	Total BG pixels	Class imbalance ratio (avg. $\pm$ std.)
BraTS [10]	1.2×10^7	253.2×10^7	712.8 ± 1463.1
Anatomical tracings of lesions after stroke (ATLAS) [152]	4.6×10^6	1908.6×10^6	1768.7 ± 6710.8
Kidney tumor segmentation (KiTS) [96]-Kidney	13.5×10^6	1662.0×10^6	122.8 ± 69.3
KiTS [96]-Tumor	2.9×10^6	1662.0×10^6	6736.6 ± 1522.9
Abdomen [132]-Gallbladder	1.0×10^5	2630.5×10^5	4887.5 ± 4854.7
Abdomen [132]-Aorta	3.8×10^5	2630.5×10^5	829.2 ± 686.1
Abdomen [132]-Vena cava	3.4×10^5	2630.5×10^5	832.2 ± 278.2
Abdomen [132]-Vein	1.4×10^5	2630.5×10^5	2196.5 ± 792.2

Class imbalance has been the focus of many previous works [27, 115]. In the following, we discuss related work categorized by different methodological approaches.

2.4.1.1 Re-weighting

A common approach to tackle class imbalance is class-level re-weighting, which assigns higher weights or higher sampling probability to the under-represented classes based on sample frequency [237, 253] or advanced rules [55]. Beyond that, sample-level re-weighting strategies are also proposed to build a balanced model. For example, hard sample mining was proposed to avoid the dominant effect of majority classes [61]. Similarly, focal loss and its variants were proposed to weight difficult samples over easy samples [1, 88, 154, 240] to steer the learning towards small objects. However, the under-represented samples are not necessarily difficult to predict during training. Margin based loss functions were proposed to learn discriminative embeddings and widely adopted to metric-learning and face recognition [56, 161].

2.4.1.2 Data synthesis

Our work is related to data synthesis methods [37, 84] which generate synthetic samples of the minority classes based on intra-class relationship between samples to increase the variance of under-represented class.

2.4.1.3 Other methods

The above mentioned methods are all based on changing the training data distribution to tackle class imbalance. In contrast, some other approaches try to counter class imbalance by modifying the training strategy. Specifically, [91] firstly trained their model with data which is sampled from each class with the same probability. Thereafter, they only retrain the output layer with uniformly sampled data while freezing all other network parameters. In this way, they could separately learn a diverse representation and a classifier for realistic data distribution. Similar strategies, which aim to change the decision boundary at test-time, are also proposed recently for long-tailed recognition [118, 226, 265]. Other learning paradigms such as meta-learning [238] and transfer learning [164] were also recently proposed for long-tail learning.

2.4.1.4 Segmentation

The problem of class imbalance in image segmentation is different from that in image recognition [164] because the dominating class in image segmentation is the background class with diverse characteristics, and its segmentation accuracy is highly robust. In contrast, the accuracy for the majority classes in long-tailed image recognition can degrade with common techniques such as re-weighting [118]. In addition, the evaluation of segmentation performance mostly relies on the foreground classes, and therefore, the focus is on improving accuracy in those classes. For example, recent studies proposed to provide a better trade-off between sensitivity and precision for segmentation [88, 175]. This is achieved by training with loss functions based on Dice similarity coefficient (DSC) but containing different weights for sensitivity and precision. However, these strategies yield little improvements when processing highly imbalanced datasets, as shown in our experiments. This is because a DNN may achieve near-perfect training accuracy even for the under-represented samples without benefitting from the modified loss function. Class imbalance in image segmentation has been also approached via a boundary loss [122]. However, it is only applicable to segmentation but cannot generalize to classification. The authors show promising results with sufficient training data, but the model may still be prone to overfit the under-represented class with limited dataset. Other work adopted multi-stage approaches with candidate proposals and background suppression [210, 231]. However, the candidate prediction process may still suffer from class imbalance. In addition, any missed candidates in one stage cannot be recovered in a later stage.

2.4.2 Domain Shifts

The success of deep learning models depends on the assumption that the learned model would be deployed with data that is drawn from the same distribution with the training data. However, in practice, $\mathcal{D}_{train}$ can be different from $\mathcal{D}_{test}$. The discrepancy between the training and test data is known as domain shift. As a data driven model, DNN cannot perform well under domain shifts. Domain shift is common issue in medical imaging as the medical images can be collected with different parameters from different clinical sets [33]. For example, we show an example of prostate MRI [159] from six different scanners in Fig. 2.11. Apart from the acquisition shift, the training dataset could be also biased because of imbalanced distributions of gender, age and ethnicity [170].

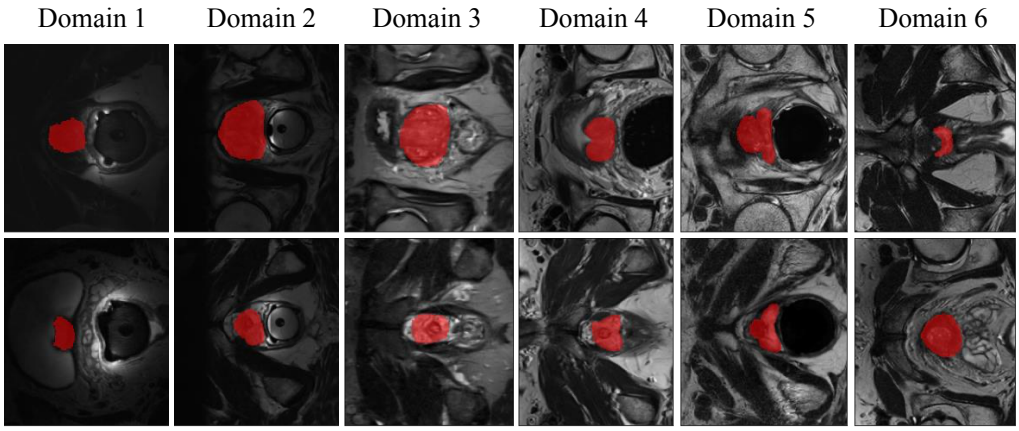


Figure 2.11: The illustration of domain shifts in medical imaging. We visualize prostate T2 weighted MR images collected with six different scanners. The ROIs are marked with red color. The appearance and contrast are different across different domains, causing difficulties for CNN when trained with data from one domain but tested on others.

Many methods have been proposed to improve cross-domain performance of DNNs. Assuming that we can access unlabeled images from the target domains, domain adaptation methods [116, 229] are proposed to align the learned features between the source and target domains. Furthermore, domain generalization [63, 267] and domain randomization [19] methods do not have any assumptions on the unseen target domains and aim to learn domain invariant features.

Although domain robustness can be improved by these techniques, it is hard to validate the model performance if we cannot access the ground truth labels in the target datasets. It is hard to notice model failure of a modern DNN as it is likely to produce over-confident predictions [83].

Performance estimation methods such as automated quality control (QC) aim to assess the prediction accuracy without the need of the ground truth labels [170]. Performance estimation without the ground truth is a valuable tool for the safe deployment of a machine learning model and also

a focus of this study. For example, the clinicians can decide whether to deploy the given learned DNN to prostate MRI from local machines based on the prediction accuracy of few unlabeled images.

Chapter 3

Overfitting under Class Imbalance

The work presented in this chapter is inspired by the findings that neural networks are more likely to under-segment small regions of interest (ROIs) with the presence of class imbalance. This chapter starts from analyzing the effects of class imbalance on model training. We aim to gain a better understanding of network behaviours with the plots of logit distribution. The work in this chapter is based on the publications:

1. **Z. Li**, K. Kamnitsas and B. Glocker. Overfitting of Neural Nets Under Class Imbalance: Analysis and Improvements for Segmentation. International Conference on Medical Image Computing and Computer-Assisted Intervention, 2019: 402-410.

2. **Z. Li**, K. Kamnitsas and B. Glocker. Analyzing Overfitting Under Class Imbalance in Neural Networks for Image Segmentation. IEEE Transactions on Medical Imaging, 2020, 40(3): 1065-1077.

Class imbalance poses a challenge for developing unbiased, accurate predictive models. In particular, in image segmentation neural networks may overfit to the foreground samples from small structures, which are often heavily under-represented in the training set, leading to poor generalization. In this study, we provide new insights on the problem of overfitting under class imbalance by inspecting the network behavior. We find empirically that when training with limited data and strong class imbalance, at test time the distribution of logit activations may shift across the decision boundary, while samples of the well-represented class seem unaffected. This bias leads to a systematic under-segmentation of small structures. This phenomenon is consistently observed for different databases, tasks and network architectures. To tackle this problem, we introduce new asymmetric variants of popular loss functions and regularization techniques including a large margin loss, focal loss, adversarial training, mixup and data augmentation, which are explicitly designed to counter logit shift of the under-represented classes. Extensive experiments are conducted on several challenging segmentation tasks. Our results demonstrate that the proposed modifications to the objective function can lead to significantly improved segmentation accuracy compared to baselines and alternative approaches.

3.1 Introduction

When machine learning model is trained with imbalanced datasets, it can overfit to the training samples from the under-represented classes and may not generalize well during test time. However, the effects of overfitting under class imbalance on the model behavior is not well understood. In this study, we investigate how the distribution of activations of the classification layer (*logits*) changes when the model is trained using different amounts of training data with strong class imbalance. As the model is trained with fewer training data and overfit the under-represented classes more, we find that the model projects unseen samples of the under-represented classes closer to and even across the decision boundary, while samples of the over-represented classes remain unaffected. This biased distribution shift leads to under-segmentation of under-represented class. Current solutions to address class imbalance or to mitigate overfitting do not explicitly consider this asymmetric logit shift and are unable to lead to significant improvements, as we show through an extensive set of experiments.

This study sheds new light on the problem of overfitting in the presence of class imbalance by making the following key contributions: 1) Via inspection of the network behavior on four segmentation tasks and datasets, and two popular model architectures, we find that overfitting under class imbalance consistently leads to decreased performance on under-represented classes specifically in terms of low sensitivity. Here, we provide a new angle on the problem from model behavior. Compared with the literature on class imbalance, a key contribution of this study is the focus on the model behaviour when it overfits to the under-represented classes with a detailed analysis and potential solutions. 2) We identify the shift in the logit distribution of unseen test samples of under-represented classes as a result of overfitting under class imbalance. 3) Based on our observations, we propose simple yet effective asymmetric variants of five loss functions and regularization techniques which are explicitly designed to change the network behavior yielding improved segmentation accuracy for the under-represented classes.

Our findings here confirm our initial observations about the biased behavior of neural networks. The behavior of logit distribution shift is consistently observed across different types of data, tasks and architectures. Our work highlights the importance of the issue of overfitting under class imbalance. The quantitative evaluation further supports our proposal of taking class imbalance into account when designing the learning objective.

3.2 Related Works

3.2.1 Class Imbalance

In this study, we explore re-weighting as a baseline approach in all experiments where we train the models with patches which are separately sampled from different classes with the same probability. In fact, as we show empirically, the training samples of the under-represented classes are learned well due to overfitting. In this case, we find that a focal loss may even decrease the performance when processing imbalanced datasets because it reduces the focus on the under-represented samples. Therefore, in this study, we improve upon focal loss by removing the attenuation of under-represented classes.

Large margin loss is a regularized loss function which assigns margins on model outputs to make the models map their predictions away from the decision boundary and therefore learn discriminative features [161]. Margin losses can be seen as a kind of re-weighting approach which changes the magnitude of the gradient of the network output by multiplying a scalar. We propose to only assign margins for the under-represented classes. The design of uneven margins for imbalanced datasets was first proposed in [143] for document categorisation problems. Recently, large margin local embedding (LMLE) was proposed to set more constraints to the under-represented classes by only applying multiple margins to the minority classes, with a computationally expensive metric-learning based framework [104]. More recently, two concurrent studies were also proposed to set larger margins for the under-represented classes from the perspective of uncertainty [123] or generalization bound [30, 268]. In this study, we empirically show that one should not assign margins for the over-represented classes based on the observations of asymmetric logit distribution under class imbalance. Different from current methods based on data synthesis, we create synthetic samples in the latent feature space rather than image space and provide two new ways to synthesize samples of the under-represented classes for modern machine learning models. We also propose to adopt stronger data augmentation for the under-represented classes by changing the augmentation probabilities to alleviate overfitting. Our methods also are complementary to solutions based on changing training strategy and could be combined. The solutions proposed here are general loss functions that can be incorporated into any model or learning approach and is potentially applicable beyond image segmentation.

3.2.2 Regularization Techniques

To improve generalization of deep neural networks, a number of regularization techniques are available. This includes dropout [222], weight decay [128], data augmentation [53, 54], data mixing [252, 261], and adversarial training [79, 244]. However, most of these techniques were proposed for general image classification tasks where class imbalance is not explicitly addressed. It is also unclear how these techniques affect the network behavior in this setting.

3.3 Analysis

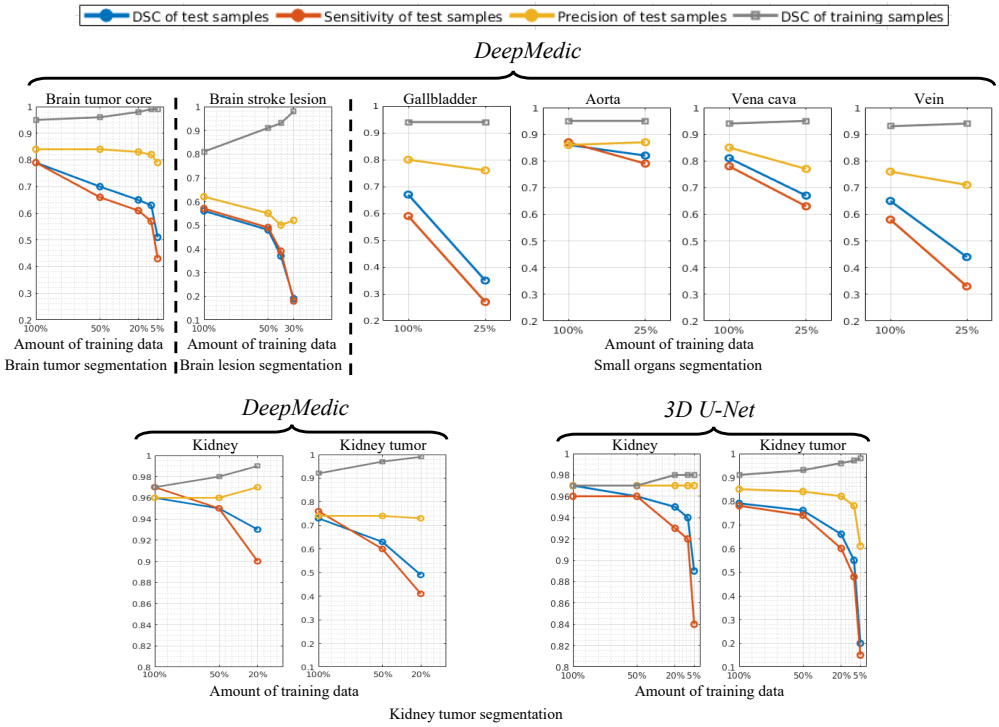


Figure 3.1: Performance on brain tumor core, brain stroke lesion, small organs, kidney and kidney tumor segmentation with varying amounts of training data. The foreground (FG) and background (BG) samples are highly imbalanced. With less training data, performance drops due to the decrease of sensitivity, while the precision is largely retained. We train DeepMedic with cross entropy (CE) while 3D U-Net with a combination of CE and Dice similarity coefficient (DSC).

3.3.1 Under-Segmentation due to Class Imbalance

To explore the effects of overfitting on the network behavior, we train convolutional neural networks (CNNs) using different amounts of data, on segmentation tasks that exhibit strong class imbalance. We conduct experiments on challenging segmentation tasks using data from the brain tumor segmentation (BraTS) challenge [10], the anatomical tracings of lesions after stroke (ATLAS) dataset [152], small organ segmentation (data from [132]) and kidney tumor segmentation (KiTS) challenge [96]. The statistics of those four datasets are summarized in Table 2.1. To ensure our findings generalize across models, in our investigation we employ two convolutional network architectures that have been proven potent in a variety of segmentation tasks: We employ a DeepMedic architecture [117] for the experiments on brain lesions and multi-organ segmentation tasks on which it has previously shown high performance [117, 208], and a three-dimensional (3D)

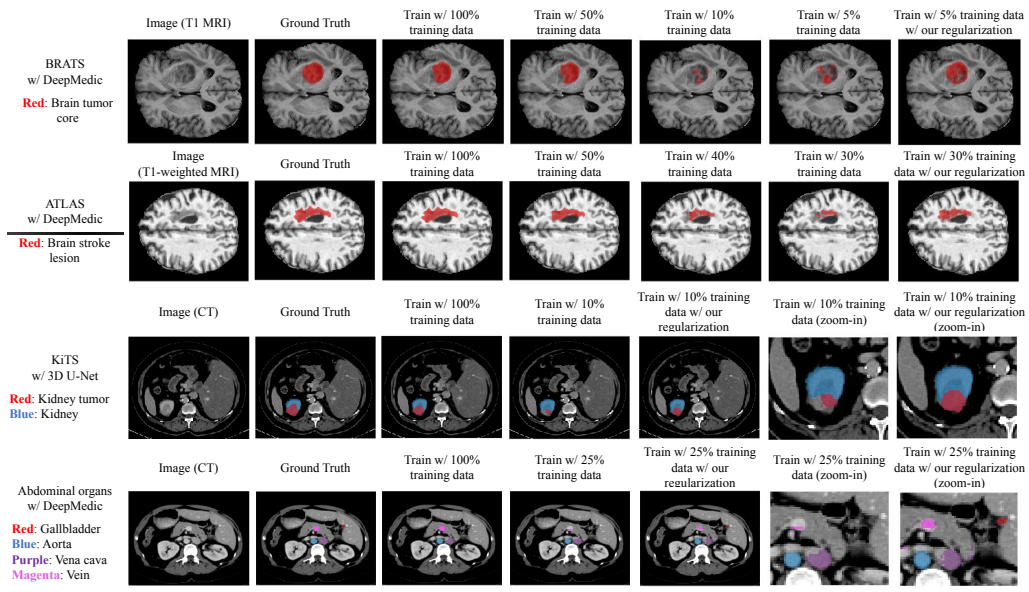


Figure 3.2: Visualization of different datasets and segmentation results with different portions of training data. With less training data, the models are prone to under-segment the under-represented classes. The proposed regulation methods can alleviate the overfitting of under-represented classes and provide segmentation results with higher sensitivity and overall accuracy. Best viewed in color.

U-Net [107] for the experiments on kidney tumor segmentation on KiTS data, which is the base model of the winning entry of KiTS challenge [96]. The detailed network configurations are summarized in Section 3.5.

The observations on the test and training set are summarized in Fig. 3.1. With less training data, we notice a clear decrease of segmentation accuracy on test data while the accuracy on training data increases due to easier overfitting, as expressed by DSC (defined as $\text{DSC} = 2\frac{\text{SEN}\cdot\text{PRC}}{\text{SEN}+\text{PRC}}$). We observe that overfitting leads to a reduction of sensitivity while precision remains largely stable. In all settings and tasks, the specificity of the foreground always remains near-perfect ($>$0.999) with different amounts of training data, indicating the predictions of background samples are stable (not shown in Fig. 3.1 to avoid cluttering). We also provide some corresponding visualization examples in Fig 3.2.

Our observations on four different datasets show that this behavior is consistent when foreground classes are under-represented and is not specific to particular tasks and model architectures. Our findings reveal that models that overfit to imbalanced training data have a bias to under-segment the under-represented class on unseen test data. We find our observations are not limited to specific loss functions as we train DeepMedic with CE while 3D U-Net with a combination of CE and DSC.

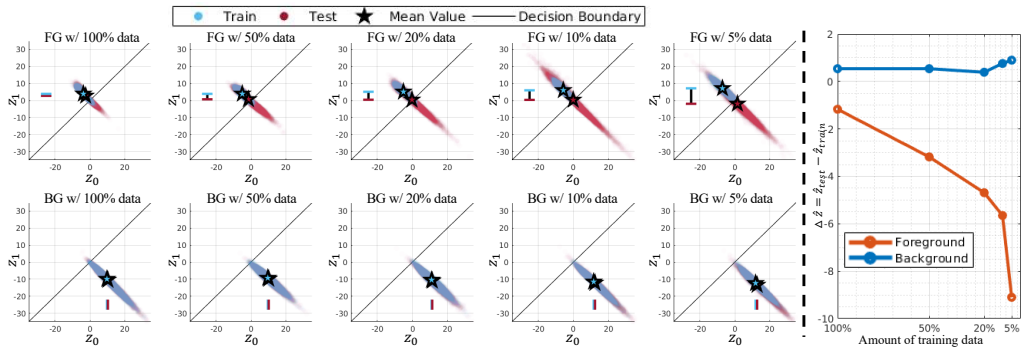


Figure 3.3: Logit distributions when processing BraTS with DeepMedic. (Left part) Activations of the classification layer (logit z_0 for background, logit z_1 for brain tumor core) when processing (top) tumor and (bottom) background samples of BraTS with DeepMedic, using different amounts of training data. The CNN maps training and testing samples of the background class to similar logit values. However, mean activation for testing data shifts significantly for the tumor class towards and sometimes across the decision boundary. (Right part) The shift of mean value of logits observed when processing training and testing data $(\Delta\hat{z} = \hat{z}_{test} - \hat{z}_{train})$.

3.3.2 Logit Distribution Shift

To obtain a better understanding of the network behavior after training on imbalanced data, we monitor the logit distribution when processing training and unseen test samples. The observations we make for the tasks of brain tumor core, kidney tumor and brain stroke lesion segmentation are summarized in Fig. 3.3, 3.4 and 3.5, respectively. We notice that the logit distribution of foreground samples shifts significantly towards and even across the decision boundary, while the logit distribution of background samples remains stable. The shift of the foreground logits results in a higher number of false negatives, which causes a drastic decrease of sensitivity (calculated as $\frac{TP}{TP+FN}$). This biased logit shift under class imbalance may also occur in other tasks such as image classification. However, it is particularly prevalent in image segmentation with small structures-of-interest.

We find that this shift of logits correlates with how much a model overfits to the under-represented class. Training with less data leads to more overfitting, and the logit distribution shift becomes larger. Moreover, we find that the logit shift also correlates with the size of structures represented by the foreground class. The rarest class shifts the most, as shown in the right part of Fig. 3.4.

In image segmentation, a CNN is optimized to push the logits of different classes away from each other and far from the decision boundary. It is relatively easy for a deep CNN to build an embedding for the training samples from the under-represented class because it just needs to build a set of case-specific filters to facilitate memorization. For example, as a CNN will only observe very

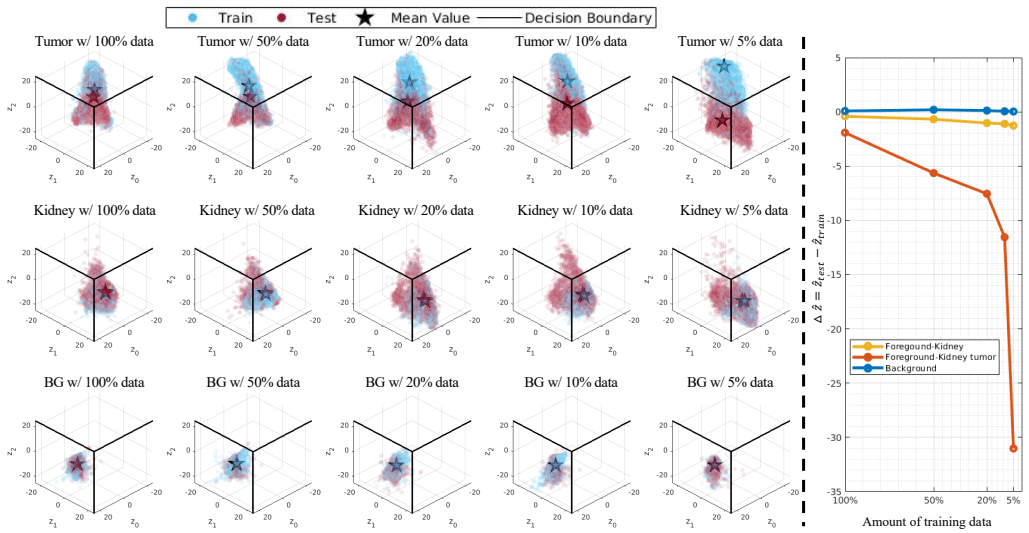


Figure 3.4: Logit distributions when processing KiTS with 3D U-Net. (Left part) Activations of the classification layer (logit z_0 for background, logit z_1 for kidney, logit z_2 for kidney tumor) when processing (top) tumor, (middle) kidney and (bottom) background samples of KiTS with 3D U-Net, using different amounts of training data. The CNN also fails to map the training and testing samples of the tumor class in a similar position. (Right part) The shift of mean value of logits.

few training samples of the foreground class, a CNN can dedicate specific model parameters to memorize all foreground samples, even if the individual patterns are rather complex. Specifically, we find a CNN seems to be more confident about foreground samples during training, mapping them farther away from the decision boundary when overfitting, as shown in Fig. 3.3, Fig. 3.4 and Fig. 3.5. However, these tailored filters will not generalize to unseen test data. The filters are optimized to have large activations when processing training data with specific patterns. The patterns of unseen samples, however, could have discrepancy from the training data. As a result, the suboptimal pattern matching of filters and unseen samples would decrease the activation magnitude. Therefore, the activations for test samples of the under-represented class are smaller in magnitude on average, leading to the observed distribution shift. In contrast, a CNN has to build generic filters for a well represented class to represent many different characteristics of the same class, leading to good generalization. Such filters will map unseen samples to similar locations in logit space and no shift between the embeddings of training and test samples is observed. As a result of class imbalance *and* overfitting, a CNN may underperform on the under-represented class while still generalizing well for the well-represented class.

While the negative effect of class imbalance and overfitting on model performance is well known, to our knowledge there has been little work investigating the specifics how the network behavior is affected. Only by understanding better the implication of class imbalance, we can devise mitigation strategies. Previous loss functions and regularization techniques that aim to prevent

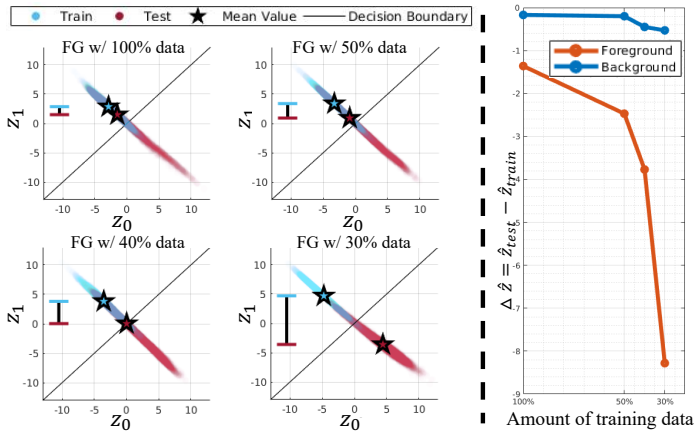


Figure 3.5: Logit distributions when processing ATLAS with DeepMedic. (Left part) Activations of the classification layer when processing lesion samples of ATLAS with DeepMedic, using different amounts of training data. (Right part) The shift of mean value of logits.

overfitting did not take the behavior that we observe into account, and thus show limited success for improving segmentation accuracy in the setting of limited data with strong class imbalance. Here, we propose solutions via new asymmetric variants of existing objective functions leading to better feature embeddings for the under-represented samples, leading to significant improvements in segmentation accuracy for small structures-of-interest.

3.4 Method

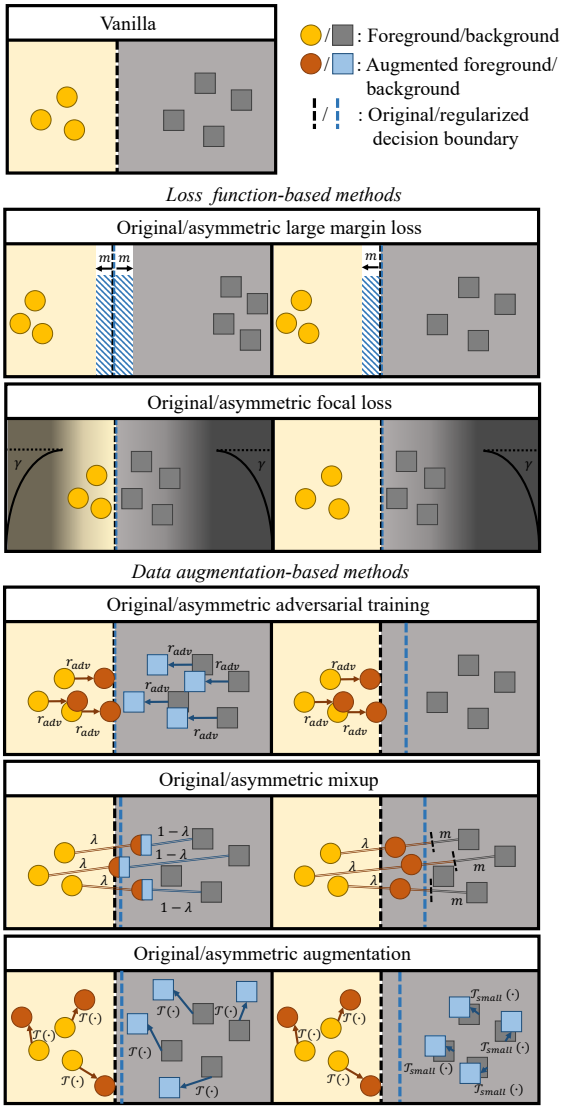


Figure 3.6: The illustration of the proposed asymmetric modifications for the existing loss functions and regularization techniques. We optimize the logit activations of foreground class to stay far away from the decision boundary by setting a bias for the foreground class in different ways.

Based on our observations above about the biased behavior of CNNs, we design modifications to existing loss functions and training strategies to prevent the logit distribution shift. Specifically, we add a bias for the under-represented class. Although the original techniques were proposed for different purposes, our modifications share a common goal: to keep the logit activations of the under-represented class away from the decision boundaries. Even if the logit of a foreground

sample shifts towards the decision boundary as long as it does not cross it, its prediction remains correct (cf. Fig. 3.6).

3.4.1 Asymmetric Large Margin Loss

We consider a CNN for the task of semantic segmentation. Following the notions in section 2.1.1, given a training dataset $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$ with N samples, we denote a training sample with $\boldsymbol{x}_i$ and its corresponding one-hot vector $\boldsymbol{y}_i$. If c is the total number of classes of the task, $\boldsymbol{y}_i$ has c elements, with its j'th element $y_{ij} \in \{0, 1\}$ corresponding to the j'th class. y_{ij} equals to 1 if j is the real class of $\boldsymbol{x}_i$, or 0 otherwise. With this notation, the CE loss can be written as[1]:

$$\mathcal{L}_{CE}(\boldsymbol{x}_i, \boldsymbol{y}_i) = -\sum_{j=1}^{c} y_{ij} \log(p_{ij}), \tag{3.1}$$

where p_{ij} is the predicted probability by the network that the real class of $\boldsymbol{x}_i$ is j. Probability p_{ij} is commonly obtained via a softmax function over the c activations $\{(z_{ij})_{j=1}^{c} \in \mathbb{R}^c\}$ that the network outputs and activations for $\boldsymbol{x}_i$ at its last layer, which we refer to *logits*. With this, p_{ij} is given by:

$$p_{ij} = \frac{\mathrm{e}^{z_{ij}}}{\sum_{j=1}^{c} \mathrm{e}^{z_{ij}}}. \tag{3.2}$$

Besides CE, the smooth version of the DSC metric is an alternative choice for the loss function which is widely used for medical image segmentation [175]. DSC loss can be calculated in the form of $1 - \text{DSC} = \frac{\text{FP}+\text{FN}}{2\,\text{TP}+\text{FP}+\text{FN}}$), which is:

$$\mathcal{L}_{DSC}(\boldsymbol{x}_i, \boldsymbol{y}_i) = \sum_{j=1}^{c} \Big(\frac{(1 - y_{ij})p_{ij} + y_{ij}(1 - p_{ij})}{y_{ij} + p_{ij}} \Big). \tag{3.3}$$

There could be other forms for DSC loss. Here, we choose this formulation for simplicity.

The large margin loss was proposed for increasing the Euclidean distances between logits for different classes to learn discriminative features [235]. Symmetrically, it is implemented by adding a margin on the logits of every class:

$$\mathcal{L}_{CE_M}(\boldsymbol{x}_i, \boldsymbol{y}_i) = -\sum_{j=1}^{c} y_{ij} \log(q_{ij}), \tag{3.4}$$

in which we require:

[1] We formulate CE as sum over classes, as opposed to sum over samples, to make class specific modifications.

$$q_{ij} = \frac{\mathrm{e}^{z_{ij} - y_{ij} m}}{\sum_{j=1}^{c} \mathrm{e}^{z_{ij} - y_{ij} m}}, \tag{3.5}$$

where m is a hyper-parameter for the margin. Although the large margin loss encourages the model to map different classes away from each other, the decision boundary remains in the center. According to our observations, class imbalance causes shifts of unseen foreground samples towards the background class. To mitigate this, a regularizer may aim to move the decision boundary closer to the background class. Our asymmetric modification only sets the margin for the rare classes. We define $\boldsymbol{r}$ as a one-hot vector with c elements, with its j'th element $r_j \in \{0, 1\}$ corresponding to the j'th class and r_j equals to 1 if j is taken as the rare class. With the indication of $\boldsymbol{r}$, we derive the asymmetric large margin loss as:

$$\hat{\mathcal{L}}_{CE_M}(\boldsymbol{x}_i, \boldsymbol{y}_i) = -\sum_{j=1}^{c} y_{ij} \log(\hat{q}_{ij}), \tag{3.6}$$

where we require:

$$\hat{q}_{ij} = \frac{\mathrm{e}^{z_{ij} - y_{ij} r_j m}}{\sum_{j=1}^{c} \mathrm{e}^{z_{ij} - y_{ij} r_j m}}, \tag{3.7}$$

In this study, we define r_j as 1 for the foreground samples and 0 for the background samples. In other applications, r_j can also be defined as a continuous variable indicating the rarity of the classes with $r_j \in [0, 1]$, for methods in Section 3.4.1, 3.4.2 and 3.4.3. Similarly, the symmetric and asymmetric large margin loss for DSC loss can be derived by substituting equation 3.5:

$$\mathcal{L}_{DSC_M}(\boldsymbol{x}_i, \boldsymbol{y}_i) = \sum_{j=1}^{c} \Big(\frac{(1 - y_{ij}) q_{ij} + y_{ij}(1 - q_{ij})}{y_{ij} + q_{ij}} \Big), \tag{3.8}$$

and

$$\hat{\mathcal{L}}_{DSC_M}(\boldsymbol{x}_i, \boldsymbol{y}_i) = \sum_{j=1}^{c} \Big(\frac{(1 - y_{ij}) \hat{q}_{ij} + y_{ij}(1 - \hat{q}_{ij})}{y_{ij} + \hat{q}_{ij}} \Big). \tag{3.9}$$

3.4.2 Asymmetric Focal Loss

The focal loss was proposed for small object detection by reducing the weight for well-classified samples and focusing on samples which are near the decision boundary [154]. It adds attenuation inside the loss function based on the logit activations:

$$\mathcal{L}_{CE_{focal}}(\boldsymbol{x}_i, \boldsymbol{y}_i) = -\sum_{j=1}^{c}(1-p_{ij})^{\gamma} y_{ij} \log(p_{ij}), \tag{3.10}$$

where γ is the hyper-parameter to control the focus. The symmetric focal loss prevents logits from being too large and makes every class stay near the decision boundary. However, this makes it likely for the unseen foreground samples to shift across the decision boundary. We remove the loss attenuation for the foreground class to keep it away from the decision boundary:

$$\hat{\mathcal{L}}_{CE_{focal}}(\boldsymbol{x}_i, \boldsymbol{y}_i) = \sum_{j=1}^{c}\Big(- r_j y_{ij} \log(p_{ij}) - (1-r_j)(1-p_{ij})^{\gamma} y_{ij} \log(p_{ij})\Big). \tag{3.11}$$

Inspired by the focal loss [154], related work integrates a similar attenuation term into the DSC loss [1, 240]. In practice, we find that the logarithmic DSC loss [240] significantly changes the magnitude of DSC loss making it difficult to be combined with other losses. The attenuation in the focal Tversky loss [1] is very large and may overly suppress the easier class. Here, we propose another form of DSC loss with an adaptive weight preserving a similar loss magnitude. Specifically, we add the attenuation term to the false negatives part of the function and prevent the network being too confident about its prediction:

$$\mathcal{L}_{DSC_{focal}}(\boldsymbol{x}_i, \boldsymbol{y}_i) = \sum_{j=1}^{c}\Big(\frac{(1-y_{ij})p_{ij} + (1-p_{ij})^{\gamma} y_{ij}(1-p_{ij})}{y_{ij} + p_{ij}}\Big), \tag{3.12}$$

Compared with the original version of the CE loss, this formulation for the DSC loss has a similar effect of reducing the penalty for the well-classified samples while keeping the magnitude of the loss similar to the original one. We refer to this as the focal DSC loss in the following. Similarly, the asymmetric version of the focal DSC loss is derived by removing the attenuation term for the foreground class:

$$\begin{aligned}\hat{\mathcal{L}}_{DSC_{focal}}(\boldsymbol{x}_i, \boldsymbol{y}_i) = \sum_{j=1}^{c}\Big(&\frac{(1-y_{ij})p_{ij} + r_j y_{ij}(1-p_{ij})}{y_{ij} + p_{ij}} \\ &+ \frac{(1-y_{ij})p_{ij} + (1-r_j)(1-p_{ij})^{\gamma} y_{ij}(1-p_{ij})}{y_{ij} + p_{ij}}\Big).\end{aligned} \tag{3.13}$$

3.4.3 Asymmetric Adversarial Training

Adversarial training was proposed to learn more robust classifiers by training with difficult samples [79] of which network predictions have large discrepancy from the ground truth. The network is trained by considering adversarial samples as additional training data [177, 244]:

$$\mathcal{L}_{adv}(\boldsymbol{x}_i, \boldsymbol{y}_i) = \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{y}_i) + \mathcal{L}(\boldsymbol{x}_i + l \cdot \frac{\boldsymbol{d_{adv}}}{\|\boldsymbol{d_{adv}}\|_2}, \boldsymbol{y}_i), \tag{3.14}$$

$$\text{with} \quad \boldsymbol{d_{adv}} = \underset{\boldsymbol{d};\|\boldsymbol{d}\|<\epsilon}{\operatorname{argmax}} \mathcal{L}(\boldsymbol{x}_i + \boldsymbol{d}, \boldsymbol{y}_i). \tag{3.15}$$

Here, $\boldsymbol{d_{adv}}$ is the direction of the generated adversarial samples, l and ϵ are the magnitude and the range of the adversarial perturbations, respectively. $\mathcal{L}$ is the chosen loss function, which can be $\mathcal{L}_{CE}$ and/or $\mathcal{L}_{DSC}$. In practice, we optimize $\boldsymbol{d_{adv}}$ with one step of gradient descend. Similar to the large margin loss, symmetric adversarial training preserves the decision boundary and may cause difficulties for unseen foreground samples, which tends to shift towards background class. Our proposed asymmetric adversarial training aims to produce a larger space between the foreground class and the decision boundary. Specifically, we generate samples by considering more from the rare classes:

$$\hat{\boldsymbol{d}}_{\boldsymbol{adv}} = \mathbb{1}_{[\boldsymbol{y}_i \cdot \boldsymbol{r} > 0]} \underset{\boldsymbol{d};\|\boldsymbol{d}\|<\epsilon}{\operatorname{argmax}} \mathcal{L}(\boldsymbol{x}_i + \boldsymbol{d}, \boldsymbol{y}_i \odot \boldsymbol{r}), \tag{3.16}$$

where "$\odot$" refers to the element product, "$\cdot$" refers to the dot product and $\mathbb{1}_{[\boldsymbol{y}_i \cdot \boldsymbol{r} > 0]} \in \{0, 1\}$ is an indicator function which is equal to 1 if and only if $\boldsymbol{y}_i \cdot \boldsymbol{r} > 0$.

3.4.4 Asymmetric Mixup

Mixup is a simple yet effective data augmentation algorithm to improve generalization by generating extra training samples by using the linear combination of pairs of images and their labels [261]:

$$\mathcal{L}_{mixup}(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{x}_k, \boldsymbol{y}_k) = \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{y}_i) + \mathcal{L}(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{y}}_i), \tag{3.17}$$

where $(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{y}}_i)$ is the generated training sample:

$$\tilde{\boldsymbol{x}}_i = \lambda \boldsymbol{x}_i + (1 - \lambda)\boldsymbol{x}_k, \quad \tilde{\boldsymbol{y}}_i = \lambda \boldsymbol{y}_i + (1 - \lambda)\boldsymbol{y}_k. \tag{3.18}$$

Here, λ is randomly selected based on a beta distribution, $(\boldsymbol{x}_k, \boldsymbol{y}_k)$ is another random training sample. Mixup regularizes the model by centering the decision boundary between classes which helps very little in our setting. Different from the original mixup, which generates samples with soft labels, our modification generates hard labels by considered augmented samples near to the foreground samples as foreground class. Asymmetric mixup can keep the decision boundary away from the foreground class and increase the area of the foreground logit distribution. This prevents unseen

under-presented samples from shifting across the decision boundary. Specifically, the mixed image $\tilde{\boldsymbol{x}}_i$ which has a certain distance from the background class, is taken as a foreground sample:

$$\hat{\tilde{\boldsymbol{y}}}_i = \begin{cases} \boldsymbol{y}_i & \text{if } \big(\lambda > m \quad \text{and} \quad \boldsymbol{y}_i \cdot \boldsymbol{r}(1 - \boldsymbol{y}_k \cdot \boldsymbol{r}) = 1\big) \quad \text{or} \quad \boldsymbol{y}_i = \boldsymbol{y}_k, \\ \boldsymbol{y}_k & \text{if } \big(1 - \lambda > m \quad \text{and} \quad \boldsymbol{y}_k \cdot \boldsymbol{r}(1 - \boldsymbol{y}_i \cdot \boldsymbol{r}) = 1\big) \quad \text{or} \quad \boldsymbol{y}_i = \boldsymbol{y}_k, \\ \mathbf{0} & \text{otherwise,} \end{cases} \tag{3.19}$$

where m is the margin to guarantee that the augmented samples are not getting too close to background samples. In practice, we do not update the model using training samples with $\hat{\tilde{\boldsymbol{y}}}_i = \mathbf{0}$. Compared with the original mixup, the proposed asymmetric variant would make the model bias towards the minority classes.

3.4.5 Asymmetric Data Augmentation

In order to extend the latent space of the foreground class, we also evaluate a simple method to compensate class imbalance by adjusting the magnitude of augmentation for different classes. Standard data augmentation methods would preserve the label $\boldsymbol{y_i}$ and adopt the same set of heuristic transformations such as scaling and rotations to the original training sample $\boldsymbol{x}_i$ for different classes. The generated training sample $\tilde{\boldsymbol{x}}_i$ can be obtained using:

$$\tilde{\boldsymbol{x}}_i = \mathcal{T}(\boldsymbol{x}_i), \tag{3.20}$$

where $\mathcal{T}$ is the chosen transformation with certain probability. When the dataset is highly imbalanced, adding more synthesized background samples is not necessary. Our simple variant of data augmentation reduces the number of transformed samples for the background classes. In this asymmetric setting, the generated sample $\tilde{\boldsymbol{x}}_i$ is obtained using:

$$\hat{\tilde{\boldsymbol{x}}}_i = \begin{cases} \mathcal{T}(\boldsymbol{x}_i) & \text{if} \quad \boldsymbol{y}_i \cdot \boldsymbol{r} = 1, \\ \mathcal{T}_{small}(\boldsymbol{x}_i) & \text{otherwise,} \end{cases} \tag{3.21}$$

where $\mathcal{T}_{small}$ is transformations with smaller probability.

3.4.6 The Combination of Asymmetric Techniques

The above-mentioned modifications would introduce more variances for the under-represented classes in the latent space or the image space by adding a bias for the foreground class from different perspectives. In practice, some or all of the techniques can be integrated into a single model to combat overfitting under class imbalance.

Specifically, during each iteration, we can first generate different sets of online augmented samples following the asymmetric adversarial training, the asymmetric mixup and the asymmetric augmentation following equation 3.16, 3.19 and 3.21, separately. The network can then be optimized using the extended training set with the loss functions combined with the asymmetric large margin loss and asymmetric focal loss:

$$\hat{\mathcal{L}}_{CE_{combine}}(\boldsymbol{x}_i, \boldsymbol{y}_i) = \sum_{j=1}^{c} \Big(- r_j y_{ij} \log(\hat{q}_{ij}) - (1 - r_j)(1 - \hat{q}_{ij})^{\gamma} y_{ij} \log(\hat{q}_{ij}) \Big). \tag{3.22}$$

A combined DSC loss can be formulated in a similar way.

3.4.7 The Analysis of Large Margin Loss from a Sample Re-weighting Perspective

Some of the proposed methods are based on two well known loss functions, noted as focal loss and large margin loss. We argue that these two loss functions can be both seen as sample-level re-weighting methods, which change the magnitude of gradient of the network output by multiplying a scalar. Specifically, focal loss would decrease the weights of well-classified samples, while large margin loss would increase the weights of all samples, especially for well-classified samples. Here, we provide an analysis of these loss functions from sample re-weighting perspective.

Following the formulation in the main text, we first consider a CNN trained using cross-entropy loss with one sample $\boldsymbol{x}_i$ and its one-hot label $\boldsymbol{y}_i$:

$$\mathcal{L}_{CE}(\boldsymbol{x}_i, \boldsymbol{y}_i) = - \sum_{j=1}^{c} y_{ij} \log(p_{ij}) = -\boldsymbol{y}_i \cdot \log(\boldsymbol{p}_i), \tag{3.23}$$

where $\boldsymbol{p}_i$ is the calculated probability, which is a normalized term from the network output $\boldsymbol{z}_i$:

$$\boldsymbol{p}_i = \frac{\mathrm{e}^{\boldsymbol{z}_i}}{\mathrm{e}^{\boldsymbol{z}_i} \cdot \mathbf{1}}. \tag{3.24}$$

Let's look at the gradient of a general loss function $\mathcal{L}(\boldsymbol{x}_i, \boldsymbol{y}_i)$ with respect to the network parameters θ:

$$\frac{\partial \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{y}_i)}{\partial \theta} = \frac{\partial \mathcal{L}(\boldsymbol{x}_i, \boldsymbol{y}_i)}{\partial \boldsymbol{z}_i} \frac{\partial \boldsymbol{z}_i}{\partial \theta}, \tag{3.25}$$

where the former term is associated with the design of loss functions and the latter is related to the network architecture. Considering a cross entropy loss $\mathcal{L}_{CE}(\boldsymbol{x}_i, \boldsymbol{y}_i)$, we can have:

$$\begin{aligned}\frac{\partial \mathcal{L}_{CE}(\boldsymbol{x}_i, \boldsymbol{y}_i)}{\partial \boldsymbol{z}_i} &= \frac{\partial \mathcal{L}_{CE}(\boldsymbol{x}_i, \boldsymbol{y}_i)}{\partial \boldsymbol{p}_i}\frac{\partial \boldsymbol{p}_i}{\partial \boldsymbol{z}_i} \\ &= -\frac{1}{\boldsymbol{p}_i \cdot \boldsymbol{y}_i}\boldsymbol{p}_i \cdot \boldsymbol{y}_i(\boldsymbol{y}_i - \boldsymbol{p}_i) = \boldsymbol{p}_i - \boldsymbol{y}_i.\end{aligned} \tag{3.26}$$

In instance-level, we can assign different weights for different samples $\boldsymbol{x}_i$ with different scalars w_i. The weights could be derive based on the frequency of samples or other heuristic rules. Typically, we multiply $\mathcal{L}_{CE}(\boldsymbol{x}_i, \boldsymbol{y}_i)$ by w_i, and the gradient after re-weighting would become:

$$\frac{\partial\Big(w_i \mathcal{L}_{CE}(\boldsymbol{x}_i, \boldsymbol{y}_i)\Big)}{\partial \boldsymbol{z}_i} = w_i(\boldsymbol{p}_i - \boldsymbol{y}_i). \tag{3.27}$$

It can be seen that re-weighting would change the gradient of the network output for different samples and make the model fit better the chosen samples, which are assigned a larger weight w_i.

Similarly, we can also derive the gradient of the focal loss as:

$$\begin{aligned}\frac{\partial\Big(\mathcal{L}_{CE_{focal}}(\boldsymbol{x}_i, \boldsymbol{y}_i)\Big)}{\partial \boldsymbol{z}_i} &= \frac{\partial\Big(\mathcal{L}_{CE_{focal}}(\boldsymbol{x}_i, \boldsymbol{y}_i)\Big)}{\partial \boldsymbol{p}_i}\frac{\partial \boldsymbol{p}_i}{\partial \boldsymbol{z}_i} \\ &= \Big((1 - \boldsymbol{p}_i \cdot \boldsymbol{y}_i)^{\gamma} - \gamma \boldsymbol{p}_i \cdot \boldsymbol{y}_i \log(\boldsymbol{p}_i \cdot \boldsymbol{y}_i)(1 - \boldsymbol{p}_i \cdot \boldsymbol{y}_i)^{\gamma - 1}\Big)(\boldsymbol{p}_i - \boldsymbol{y}_i) \\ &= w_{i_{focal}}(\boldsymbol{p}_i - \boldsymbol{y}_i),\end{aligned} \tag{3.28}$$

The weight term of focal loss $w_{i_{focal}}$ is a scalar and related to the sample probability $\boldsymbol{p}_i \cdot \boldsymbol{y}_i$. Generally speaking, $w_{i_{focal}}$ would decrease when $\boldsymbol{p}_i \cdot \boldsymbol{y}_i$ is large, therefore focal loss would make the model fit the easy cases less.

More interestingly, we next look into the effect of large margin loss on the gradient. Large margin loss would change the calculation of probability for the training process. Specifically, we substitute $\boldsymbol{p}_i$ with $\boldsymbol{q}_i$ to calculate the loss function, where we require:

$$\boldsymbol{q}_i = \frac{\mathrm{e}^{\boldsymbol{z}_i - \boldsymbol{y}_i m}}{\mathrm{e}^{\boldsymbol{z}_i - \boldsymbol{y}_i m} \cdot \mathbf{1}}, \tag{3.29}$$

where m is the hyper-parameter for the margin. In this case, the gradient of large margin loss can be derived as:

$$\begin{aligned}\frac{\partial \mathcal{L}_{CE_M}(\boldsymbol{x}_i, \boldsymbol{y}_i)}{\partial \boldsymbol{z}_i} &= \frac{\partial \mathcal{L}_{CE}(\boldsymbol{x}_i, \boldsymbol{y}_i)}{\partial \boldsymbol{q}_i}\frac{\partial \boldsymbol{q}_i}{\partial \boldsymbol{z}_i} \\ &= \frac{\mathrm{e}^{\boldsymbol{z}_i} \cdot \mathbf{1}}{\mathrm{e}^{\boldsymbol{z}_i - \boldsymbol{y}_i m} \cdot \mathbf{1}}(\boldsymbol{p}_i - \boldsymbol{y}_i) = w_{i_M}(\boldsymbol{p}_i - \boldsymbol{y}_i).\end{aligned} \tag{3.30}$$

The weight term of large margin loss w_{i_M} is also a scalar and related to the network output $\boldsymbol{z}_i \cdot \boldsymbol{y}_i$. It can be seen that the existence of a margin m would increase the gradient of sample $\boldsymbol{x}_i$. Moreover, w_{i_M} would be larger as $\boldsymbol{z}_i \cdot \boldsymbol{y}_i$ becomes larger, therefore large margin loss would make the model fit the easy cases more, and keep the distribution of $\boldsymbol{z}_i$ away from the decision boundary.

The analysis for $\mathcal{L}_{DSC}$ can be done in a similar way.

3.5 Experiments

3.5.1 Experimental Setup

We demonstrate the effect of our proposed modifications with a variety of medical image segmentation tasks using different models and training scenarios. Here, we summarize the dataset splits and experimental settings, which are the same as motivational experiments in Section 3.3. We keep the hyper-parameters of the methods the same for the original baselines and our modified techniques.

Table 3.1: Evaluation of brain tumor core segmentation using DeepMedic with different amounts of training data and different techniques to counter overfitting. The results are calculated with post-processing. Results which have worse DSC than the vanilla baseline are highlighted with shading. The best and second best results are in **bold** with the best also **underlined**.

Method	5% training				10% training				20% training				50% training			
	DSC	SEN	PRC	HD	DSC	SEN	PRC	HD	DSC	SEN	PRC	HD	DSC	SEN	PRC	HD
Vanilla - CE [117]	50.4	41.0	83.5	18.0	62.5	56.0	83.1	14.3	64.9	59.8	85.7	13.8	69.4	65.4	85.3	15.7
Vanilla - CE - 80% tumor	45.5	36.0	86.7	17.8	61.5	54.2	81.7	18.5	65.3	59.6	85.0	15.1	68.6	64.1	86.1	14.8
Vanilla - F1 (DSC)	47.2	37.4	86.6	15.9	58.9	51.1	83.6	20.1	64.3	58.1	83.5	16.3	67.1	62.5	86.5	15.3
Vanilla - F2 [88]	45.8	36.9	81.9	17.9	59.3	52.2	84.9	18.0	66.4	61.1	83.4	14.1	68.8	66.0	83.4	13.7
Vanilla - F4 [88]	51.6	42.5	83.8	18.1	59.6	53.0	82.9	18.4	65.9	61.9	85.4	14.2	67.5	64.5	84.9	13.7
Vanilla - F8 [88]	47.4	38.7	83.1	19.6	59.8	52.4	87.0	15.4	64.5	60.3	85.2	14.7	67.9	65.4	81.6	14.9
Large margin loss [161]	44.5	35.9	82.8	20.2	60.9	53.5	84.0	17.6	67.0	61.6	86.1	14.4	66.5	62.2	88.1	13.7
Asymmetric large margin loss	56.8	48.9	83.4	**15.0**	64.0	56.8	87.0	13.9	67.4	62.9	84.1	15.9	68.9	64.9	86.5	14.1
Focal loss [154]	54.0	44.8	82.6	16.0	62.6	55.1	84.3	17.7	64.9	60.0	84.4	19.5	67.0	62.3	87.0	16.5
Asymmetric focal loss	58.8	51.4	81.6	**15.0**	66.8	62.0	83.2	**13.2**	68.9	66.2	83.3	**12.5**	**71.5**	70.6	83.7	12.1
Adversarial training [79]	53.2	44.6	85.0	19.2	62.0	55.0	84.8	20.6	64.6	59.4	84.6	17.3	65.6	61.2	86.0	19.4
Asymmetric adversarial training	58.5	50.8	80.1	16.2	63.9	58.2	83.1	17.2	67.7	63.7	84.2	17.0	70.5	68.4	83.0	14.8
Mixup [261]	49.7	40.9	83.0	19.6	60.3	53.9	83.1	21.2	63.9	58.5	84.1	18.2	66.4	61.5	86.8	19.0
Asymmetric mixup	**59.8**	56.8	74.7	17.7	**68.5**	65.1	80.7	15.3	**70.8**	67.9	85.1	**11.6**	70.7	67.9	85.4	**11.8**
Symmetric combination	50.0	42.0	84.6	21.1	60.3	53.1	84.7	25.1	64.1	58.3	86.6	19.1	67.2	63.1	86.6	15.1
Asymmetric combination	**63.4**	63.1	75.9	**15.1**	**72.4**	72.9	78.3	**10.8**	**71.6**	72.0	80.1	13.7	**74.1**	76.0	82.4	**10.7**

3.5.1.1 Brain tumor segmentation

We first evaluate the asymmetric techniques for the case of binary brain tumor core segmentation using the DeepMedic network architecture, a well performing method for this task [117]. To investigate the behavior under overfitting and to isolate better the effect of the objective functions, we do not use dropout, weight decay or data augmentation in this experiment. We train the network with CE loss, unless otherwise specified. By default, we sample 50% training samples from the foreground class. We conduct experiments using the training dataset of BraTS2017 dataset [10] which contains 285 four modalities magnetic resonance (MR) images. The MR images all have the same voxel space of $1.0 \times 1.0 \times 1.0$ mm. We test on 95 cases and train separate models using 190

Table 3.2: Evaluation of brain stroke lesion segmentation on ATLAS based on DeepMedic with different amounts of training data and different techniques to counter overfitting. The results are calculated with post-processing. Results which have worse DSC than the vanilla baseline are highlighted with shading. The best and second best results are in **bold** with the best also **underlined**.

Method	30% training				50% training				100% training			
	DSC	SEN	PRC	HD	DSC	SEN	PRC	HD	DSC	SEN	PRC	HD
Vanilla - w/ augmentation [117]	22.2	18.3	60.9	48.6	45.2	40.9	59.7	31.1	54.5	49.5	67.3	32.2
Vanilla - w/o augmentation	15.0	11.7	59.1	51.9	40.3	35.9	53.0	40.2	51.7	48.0	62.3	31.9
Vanilla - asymmetric augmentation	22.4	18.8	58.0	50.2	47.3	43.4	57.5	32.1	56.9	51.9	69.8	28.2
Large margin loss [161]	18.9	14.8	64.4	48.8	45.3	40.7	60.5	36.8	55.1	49.4	70.0	28.0
Asymmetric large margin loss	23.5	19.8	58.6	45.9	47.7	44.3	58.4	33.8	**57.6**	54.1	67.6	**27.8**
Focal loss [154]	20.4	16.7	62.7	47.9	46.9	41.8	61.7	31.4	56.0	50.8	69.1	30.9
Asymmetric focal loss	26.3	22.2	59.0	46.4	49.0	47.8	56.3	31.7	56.6	63.2	55.6	27.9
Adversarial training [79]	20.1	16.7	57.3	56.9	47.2	41.6	62.6	35.0	54.0	48.5	69.7	34.9
Asymmetric adversarial training	**28.1**	23.3	63.9	**43.4**	**50.2**	46.5	64.2	**29.6**	55.5	51.6	69.5	33.6
Mixup [261]	14.5	12.0	52.7	54.1	45.7	41.7	59.5	30.3	53.8	50.0	67.8	29.2
Asymmetric mixup	22.8	20.9	50.9	45.9	49.0	49.2	53.9	31.5	57.0	51.0	74.6	31.7
Symmetric combination	22.2	17.7	66.1	49.4	48.5	44.5	61.0	31.6	56.0	50.4	71.5	29.9
Asymmetric combination	**31.1**	27.9	57.6	**42.1**	**52.2**	50.6	59.5	**27.8**	**58.5**	54.9	70.9	**27.3**

(100%), 95 (50%), 38 (20%), 19 (10%) and 10 cases (5% of full training set).

3.5.1.2 Brain stroke lesion segmentation

We also evaluate the asymmetric techniques for the case of brain stroke lesion segmentation [152] again using DeepMedic. Here, we use a more realistic setting, employing standard regularization techniques including dropout, weight decay and data augmentation, as in the original work where the model achieved high performance for stroke lesion segmentation [117]. We implement our asymmetric techniques with the default training setting and default network architecture. The augmentation includes small intensity shifts and flipping in the sagittal plane with probability 0.5. The network is always trained with CE loss. We conduct experiments using ATLAS dataset [152] which contains 220 T1-weighted MR images. The MR images have the same voxel space of $1.0\times1.0\times1.0$ mm. We test on 75 cases and train separate models using 145 (100%), 73 (50%), 57 (40%) and 43 cases (30% of full training set).

3.5.1.3 Small organ segmentation

For organ segmentation in Section 3.3, we use a default DeepMedic network. We conduct experiments using the training datatset of the abdominal organ segmentation challenge [132] which contains 30 computerised tomography (CT) scans. We train the network to segment thirteen abdominal organs. We test on 10 cases and train models using 20 (100%) and 5 cases (25% of training set). We resample all the MR images to a common voxel spacing of $2.0\times2.0\times2.0$ mm. We show

Table 3.3: Evaluation of kidney and kidney tumor segmentation based on 3D U-Net with different amounts of training data and different techniques to counter overfitting. The results are calculated with post-processing. Results which have worse DSC than the vanilla baseline are highlighted with shading. The best and second best results are in **bold** with the best also **underlined**.

Method	Kidney											
	10% training				50% training				100% training			
	DSC	SEN	PRC	HD	DSC	SEN	PRC	HD	DSC	SEN	PRC	HD
Vanilla - w/ augmentation [107]	93.3	91.2	96.9	5.4	96.4	95.8	97.1	2.7	96.6	96.1	97.3	2.4
Vanilla - w/o augmentation	92.3	89.3	96.8	12.1	96.1	95.6	96.7	2.8	96.3	95.8	96.9	2.7
Vanilla - asymmetric augmentation	94.3	92.2	97.0	5.2	94.9	94.5	95.5	5.9	96.1	95.8	96.4	3.8
Large margin loss [161]	**94.6**	92.7	97.1	4.8	96.4	95.9	97.0	2.8	96.1	95.9	96.3	3.2
Asymmetric large margin loss	93.8	91.4	97.2	5.3	96.1	95.5	96.9	2.9	**96.8**	96.6	97.1	**2.2**
Focal loss [154]	91.4	85.9	99.2	10.6	94.1	89.6	99.2	4.2	94.3	90.0	99.1	4.2
Asymmetric focal loss	92.0	86.7	99.0	6.0	94.7	90.9	98.9	3.5	94.8	90.9	99.1	3.1
Adversarial training [79]	94.1	91.9	97.3	9.1	96.3	95.7	97.1	2.6	96.6	96.2	97.2	**2.3**
Asymmetric adversarial training	94.4	92.5	97.2	5.7	**96.6**	96.0	97.3	**2.5**	**96.8**	96.4	97.3	**2.3**
Mixup [261]	**95.0**	93.2	97.3	**4.2**	**96.8**	96.2	97.5	**2.3**	**96.9**	96.4	97.5	**2.2**
Asymmetric mixup	**94.6**	92.6	97.3	**4.5**	96.0	95.2	97.0	3.1	96.4	95.7	97.3	2.7
Symmetric combination	94.1	91.4	97.5	5.1	94.6	91.0	98.7	4.3	96.7	96.2	97.2	**2.2**
Asymmetric combination	93.5	89.7	98.5	5.2	93.9	90.0	98.3	5.3	96.7	95.6	97.9	**2.2**
Method	Kidney tumor											
	10% training				50% training				100% training			
	DSC	SEN	PRC	HD	DSC	SEN	PRC	HD	DSC	SEN	PRC	HD
Vanilla - w/ augmentation [107]	54.6	46.0	80.0	**53.2**	76.0	72.8	86.1	25.1	79.2	77.0	86.2	**17.8**
Vanilla - w/o augmentation	37.4	31.5	65.6	96.0	62.8	58.7	75.9	47.8	73.0	69.1	83.4	18.9
Vanilla - asymmetric augmentation	55.9	48.2	76.4	71.5	74.3	70.3	85.2	33.3	78.4	76.9	85.7	19.8
Large margin loss [161]	52.2	44.3	77.2	68.5	78.2	74.3	87.8	26.6	80.2	79.1	84.5	25.5
Asymmetric large margin loss	55.5	48.3	77.4	71.6	**78.4**	74.9	87.5	24.1	**82.3**	81.4	86.0	**16.9**
Focal loss [154]	47.1	37.5	78.2	74.5	73.0	66.0	87.6	40.2	79.0	73.2	90.0	20.3
Asymmetric focal loss	**57.9**	48.9	78.4	61.4	77.4	74.4	85.0	**20.2**	81.5	80.6	86.7	19.4
Adversarial training [79]	50.9	42.5	81.3	62.0	73.2	69.6	83.9	44.1	81.9	81.1	85.8	27.6
Asymmetric adversarial training	55.2	47.8	79.6	66.7	78.3	74.9	87.9	23.7	82.1	81.1	87.4	19.7
Mixup [261]	53.3	45.2	81.6	57.8	77.0	72.9	87.3	32.1	80.3	78.5	85.9	34.1
Asymmetric mixup	56.8	48.1	84.6	66.5	77.9	74.0	89.2	22.0	79.7	78.1	87.3	19.3
Symmetric combination	53.9	45.1	81.3	70.2	73.9	67.1	87.7	39.6	80.9	79.3	86.5	19.6
Asymmetric combination	**59.2**	52.2	80.3	**49.5**	**79.4**	77.0	86.7	**15.5**	**82.7**	82.1	87.0	18.8

the segmentation results of representative small organs including gallbladder, aorta, inferior vena cava as well as portal vein and splenic vein in Fig. 3.1. Here, we use this dataset to further confirm the observations about the effect on sensitivity when training with varying amounts data.

3.5.1.4 Kidney tumor segmentation

In addition, we evaluate the asymmetric techniques for the case of kidney tumor segmentation. We train a well configured 3D U-Net [107] which includes extensive data augmentation with scaling, rotations, brightness, contrast, gamma and Gaussian noise augmentations with a predefined policy [96]. We also train DeepMedic with similar augmentation strategies yielding lower accuracy on this task. Therefore, we evaluate the asymmetric regularization techniques on the U-Net with

kidney tumor segmentation. The task includes the segmentation of both the kidney and kidney tumor. As the segmentation of kidney is relatively easy, in this experiment we only focus on tumor segmentation and only take kidney tumor as the foreground class to implement asymmetric techniques. To be specific, we always set $\boldsymbol{r}$ as $[0, 0, 1]^\intercal$. The network is always trained with both CE and sample-wise DSC loss. The two losses have the same weight. We conduct experiments using the training dataset of KiTS19 dataset [96] which contains 210 CT images. We resample all the CT images to a common voxel spacing of $1.6 \times 1.6 \times 3.2$ mm following [107]. The original and asymmetric techniques are implemented on both loss functions with the same hyper-parameters. We test on 70 cases and train separate models using 140 (100%), 70 (50%), 28 (20%), 14 (10%) and 7 cases (5% of full training set). Note, we do not manually clean the training dataset or employ deep supervision as done in [107], therefore the absolute performance of our U-Net is not directly comparable with the reported challenge results.

3.5.2 Quantitative Results

Taking the provided manual segmentations as the ground truth, we calculate DSC, SEN, PRC and 95% Hausdorff distance (HD) (mm) to evaluate the segmentation accuracy. The initial segmentation results of our method always have higher sensitivity and DSC, but sometimes would cause more false positive predictions and therefore lead to worse distance-based metrics such as HD. We argue that in practice this problem can be addressed by taking advantage of some connected component-based post-processing, which is widely adopted in many segmentation methods [107]. Specifically, we assume there is only one target component and suppress all but the largest region. We report results with these post-processing operations. The quantitative segmentation results on BraTS, ATLAS and KiTS datasets using different amounts of training data with post-processing are summarized in Table 3.1, 3.2 and 3.3, respectively. We also evaluate one of the proposed methods, asymmetric focal loss, with abdominal organ segmentation in which multiple classes are considered under-represented.

Class imbalance affects the segmentation sensitivity of the under-represented class, as shown in Section 3.3. We find that previous attempts to tackle class imbalance do not improve sensitivity, while our asymmetric methods do lead to better results with higher sensitivity across different tasks. This indicates that the proposed methods may effectively mitigate overfitting under class imbalance. These results in turn support our previous analysis of logit shift under class imbalance and indicate that considering this implication would help build unbiased network.

3.5.2.1 Baseline experiments

We perform baseline experiments with binary brain tumor core segmentation, as shown in Table 3.1. We show that increasing the weight of tumor samples (from 50% to 80%) decreases performance when the dataset is highly imbalanced. This is because increasing the weight en-

courages the network to memorize the under-represented samples and may actually lead to more overfitting, thus being counter-productive. Simply changing the objective function to F-score (defined as $F_\beta = (1+\beta^2)\frac{\text{SEN}\cdot\text{PRC}}{\beta^2\text{SEN}+\text{PRC}}$), which is a balancing loss and weights sensitivity β-times more than precision [88, 175], only shows little improvements increasing the sensitivity slightly. Changing the sampling weights or training with a loss function using F-scores seems to have little impact when foreground training samples are limited, with training accuracy close to 100%, as shown in Fig. 3.1.

A common approach to alleviate under-segmentation is to adjust thresholds of decision boundaries based on validation sets. In this work, however, we observe distribution shift on unseen test data, which can significantly differ from the training/validation sets. Hence, a threshold selected on validation data may not be optimal for new test data. Due to the lack of ground truth, it is practically not possible to optimise the decision thresholds for a specific test set. As one of the future works, we might consider all thresholds with a receiver operating characteristic (ROC) curve.

3.5.2.2 Asymmetric large margin loss

The original large margin loss decreases performance in some cases, while our modification yields improvements over the symmetric version in all cases.

3.5.2.3 Asymmetric focal loss

The original focal loss also decreases the sensitivity in some cases and leads to worse performance. It is because the focal term would decrease the weight of foreground samples and push its logit closer to the decision boundary, making it easier to cause false negative predictions. Our modification removes the loss attenuation for the under-represented class and improve the performance in all cases. We notice that the asymmetric focal loss would make the performance of other class (kidney) overfit more, but it is not the focus of this study and can be easily addressed by just keeping focal term for the background class.

3.5.2.4 Asymmetric adversarial training

When the network is trained without data augmentation (as shown in Table 2.1), the original adversarial training seems to be effective when little training data is available while our modifications can further improve the sensitivity and boost the performance substantially.

When the network is trained with data augmentation (as shown in Table 3.2 and 3.3), we find the original adversarial training does not improve the performance when training data is limited. It indicates that in this case the augmented samples by adversarial training might not add anything on top of intensity augmentation. In contrast, our proposed modifications seem to help always leading to better segmentation performance.

3.5.2.5 Asymmetric mixup

We find the original mixup can be effective for the well-represented class. For example, it always improve the segmentation performance of kidney, as shown in Table 3.3. However, the original mixup leads to lower sensitivity for the under-represented class. We find that the asymmetric mixup can improve the performance for BraTS to a large extent, as shown in Table 3.1. However, we also find it to be less effective for ATLAS and KiTS which only have one image channel, as shown in Table 3.2 and 3.3. This may be because the intensity distributions of healthy and lesion regions in ATLAS and KiTS overlap too much. In this case, the mixed samples, which are very similar to the background samples in intensity but are taken as the foreground samples, may confuse the network. For BraTS, however, four image channels are available with the healthy and tumor regions show larger differences in T2 and fluid-attenuated inversion recovery (FLAIR) sequences. The mixed samples seem to take good advantage of the intensity relationship.

3.5.2.6 Asymmetric augmentation

Despite its simplicity, we find asymmetric augmentation to be an effective method to improve segmentation performance of the under-represented classes in most cases in terms of DSC and sensitivity, as summarized in Table 3.2. However, we also notice that it could decrease the performance when data augmentation is strong and training data is sufficient, as shown in Table 3.3. It might be because the strong asymmetric augmentation would drive the model to focus too much on the foreground samples, making the background samples under-represented. We could address this by optimizing another meta-learner to choose the optimal data augmentation strategies, as we will discuss in Chapter 5.

3.5.2.7 The combination of asymmetric techniques

We also combine the asymmetric techniques which are found to improve the segmentation accuracy. The combination of the asymmetric techniques is a safe choice leading to the overall best segmentation results with improved sensitivity in all cases. In contrast, the combination of the symmetric counterparts is unable to mitigate overfitting and often decreases sensitivity.

3.5.3 Logit Distribution Changes

The effects of all the techniques on the logit distributions of BraTS using 5% training data is presented in Fig. 3.7. Asymmetric techniques would increase the variances of the foreground class and expand its logit distribution. The original large margin loss and adversarial training try to push samples from different classes far from each other, however, the logits of unseen data remain in the center around the decision boundary and thus the predictions are not improved. The original large margin loss results in even larger shifts for the foreground samples. For our asymmetric modifications only the logits of foreground samples are pushed away and the unseen foreground logits

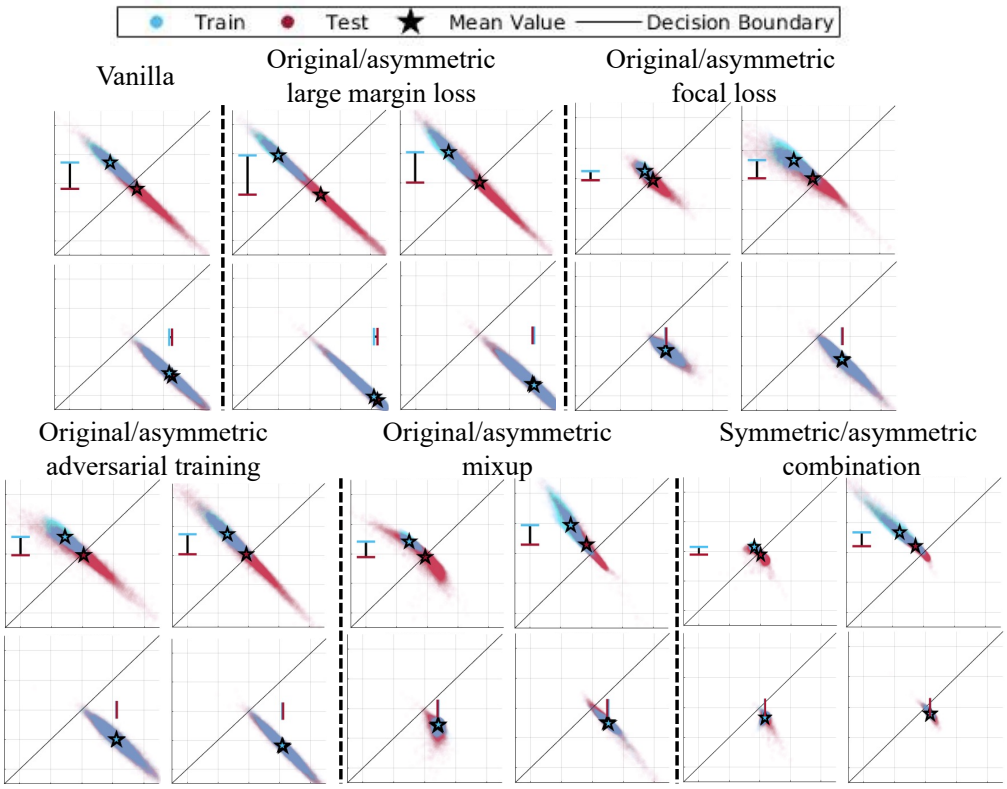


Figure 3.7: Activations of the classification layer when processing tumor (top) and background (bottom) samples of BraTS with DeepMedic, using 5% training data. Asymmetric modifications lead to better separation of the logits of unseen tumor samples.

tend to remain on the correct side of the decision boundary. The original focal loss encourages the network to prevent the logits of each class from staying too far from the decision boundary. However, it allows foreground logits to remain near the decision boundary which can result in false negative predictions on unseen samples. Our asymmetric focal loss removes the constraints for foreground samples. Original mixup encourages the symmetric distributions of different classes but does not consider class imbalance. Asymmetric mixup exploits the latent space based on the relationship between samples to generate foreground samples and make the decision boundary stay near the background class. This leads to the largest improvement by increasing the region for the foreground logit distribution and reduce logit shift of unseen foreground samples. The combination of the four asymmetric techniques can stabilize the logits further more.

The effects of all the techniques on the logit distribution of KiTS using 10% training data is summarized in Fig. 3.8. The original data augmentation can reduce the logit shift of both unseen kidney and kidney tumor samples although it is not specifically designed to regularize the logit distribution. The asymmetric augmentation can further reduce the logit shift of the unseen tumor samples. The asymmetric large margin loss reduces the tumor logit shift towards the kidney class. Although

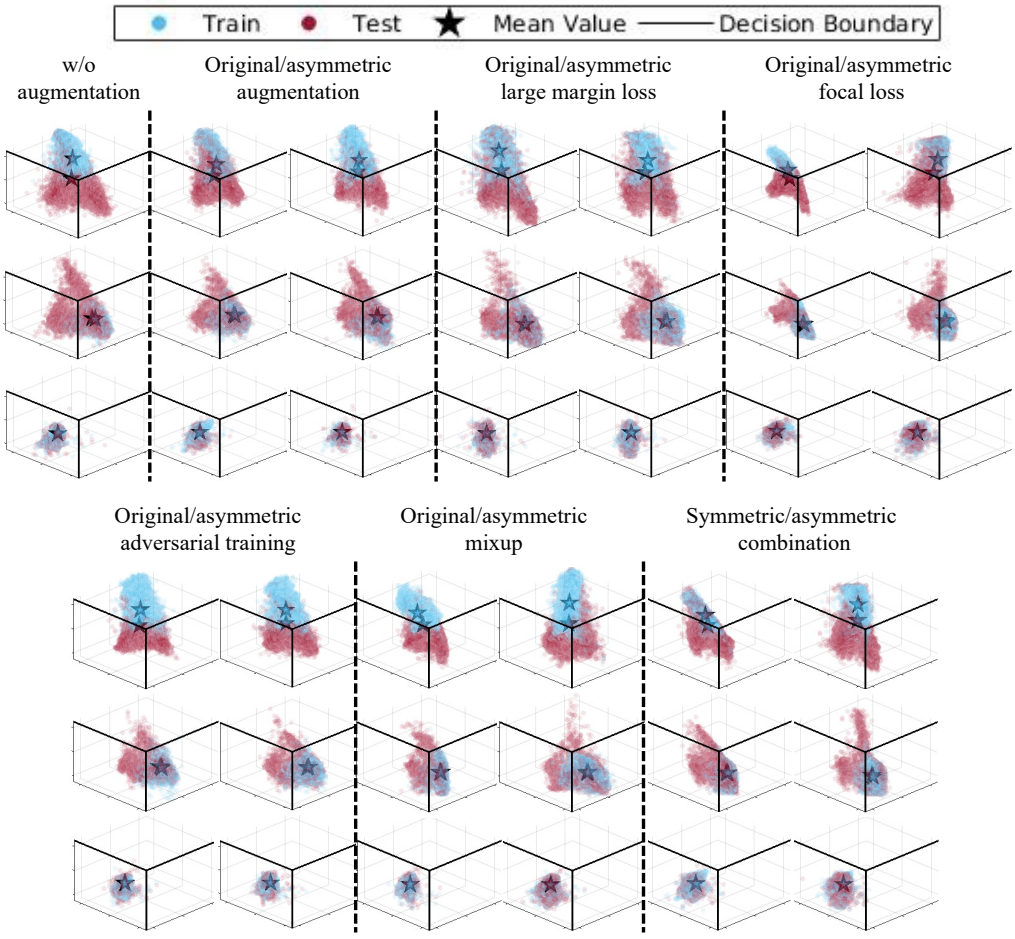


Figure 3.8: Activations of the classification layer for tumor (top), kidney (middle) and background (bottom) samples of KiTS with 3D U-Net, using 10% training data. Asymmetric modifications also lead to better separation of the logits of unseen tumor samples and is complementary to standard data augmentation.

the logit distribution is already regularized by the strong augmentation, the proposed asymmetric techniques provide further benefits in stabilizing the logits.

3.5.4 Sensitivity Analysis

We conduct a series of controlled experiments with different hyper-parameters to provide more practical details of the proposed regularization techniques. Specifically, we use a baseline DeepMedic model for brain tumor core segmentation with 5% training data of BraTS. The experimental details are consistent with descriptions in Section 3.5. We summarize the results with post-processing in Table 3.4. We can see from the results that the proposed methods can improve the baseline segmentation results with varied hyper-parameters in most cases. Specifically, asymmetric large

margin loss yields improvements for most cases, however, a specific hyper-parameter may yield unexpected results (i.e. $m = 0.5$). A potential reason is that the model which focuses on a small portion of easy under-represented samples (c.f. Eq. 3.30) would overfit more. Asymmetric large margin loss with larger m makes the model emphasize on more under-represented samples and therefore generalize better. Asymmetric adversarial training and asymmetric mixup yields considerable improvements when the perturbation in data augmentation is large (i.e. $l > 2.5$ for asymmetric adversarial training and $m < 0.8$ for asymmetric mixup). Asymmetric focal loss is robust and can improve the segmentation results with all chosen hyper-parameters. Therefore, we recommend to choose asymmetric focal loss at first for new applications.

Table 3.4: The sensitivity analysis of different hyper-parameters. We conduct experiments with different parameters with brain tumor core segmentation (5% training data) with BraTS using DeepMedic. The results are calcualted with post-processing. Results which have worse DSC than the vanilla baseline are highlighted with gray shading.

Method	Parameter		DSC	SEN	PRC	HD
Vanilla - CE	−−		50.4	41.0	83.5	18.0
Asymmetric large margin loss	m = 0.2		53.6	44.8	84.8	15.6
	m = 0.5		48.4	39.4	81.5	16.9
	m = 1		56.8	48.9	83.4	15.0
	m = 1.5		54.1	45.6	81.7	15.3
	m = 2		51.6	42.8	84.0	16.7
	m = 3		54.4	45.7	82.3	14.3
Asymmetric focal loss	γ = 0.5		53.4	44.8	79.6	16.6
	γ = 1		53.9	45.2	81.9	17.9
	γ = 1.5		56.5	48.3	87.8	13.8
	γ = 2		58.8	51.4	81.6	15.0
	γ = 3		57.5	49.0	85.5	14.2
	γ = 4		55.2	48.5	78.3	15.5
Asymmetric adversarial training	ϵ = 1e-5	l = 2.5	50.3	41.8	82.0	17.2
	ϵ = 1e-5	l = 5	58.1	50.0	84.7	14.1
	ϵ = 1e-5	l = 10	58.5	50.8	80.1	16.2
	ϵ = 1e-5	l = 15	53.8	46.2	80.1	16.9
	ϵ = 1e-5	l = 20	56.6	50.7	76.9	18.8
	ϵ = 1e-4	l = 10	57.6	51.1	78.9	16.1
	ϵ = 1e-6	l = 10	56.2	48.5	81.1	17.8
Asymmetric mixup	m = 0.1		52.1	47.3	73.8	20.7
	m = 0.15		58.1	53.7	75.0	19.9
	m = 0.2		59.8	56.8	74.7	17.7
	m = 0.25		60.4	55.0	82.0	15.6
	m = 0.3		59.1	54.3	82.0	15.3
	m = 0.4		52.1	44.2	84.2	21.4
	m = 0.8		50.3	41.6	85.5	17.7

3.5.5 The Intensity Histograms of Different Datasets

Empirically, we find the asymmetric mixup is the most effective method for tumor segmentation with BraTS. However, asymmetric mixup show limited improvements for ATLAS and KiTS. We think it is because the multi-channel information in BraTS could create more useful information, as shown in Figure 3.9.

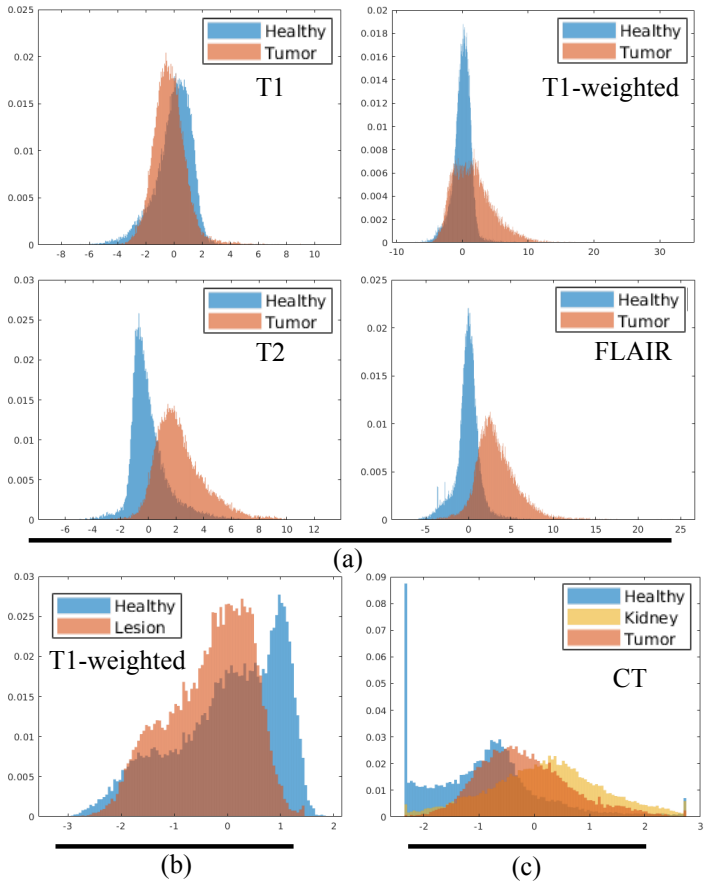


Figure 3.9: The intensity histogram of different datasets. (a) The intensity histogram of BraTS, (b) ATLAS and (c) KiTS. The intensity of the foreground and background classes overlap a lot for ATLAS and KiTS. This can be a potential factor due to which the asymmetric mixup does not create useful synthetic samples and cannot improve the segmentation performance that much.

3.5.6 Quantitative Results of Abdominal Organ Segmentation

We evaluate one of our proposed techniques, asymmetric focal loss, with the application of abdominal organ segmentation to demonstrate our method can be feasibly applied to multi-class segmentation. Specifically, we train a model of basic DeepMedic using 25% of the training data, with the same settings in empirical experiments in Section 3.3. Considering the class distribution of the dataset, as shown in Fig. 3.10, we take class 4 (gallbladder), class 5 (esophagus), class 8 (aorta), class 9 (vena cava), class 10 (vein), class 11 (pancreas), class 12 (right adrenal) and class 13 (left adrenal) as rare classes. Specifically, we initiate the one-hot vector $\boldsymbol{r}$ as $[0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1]^{\intercal}$. We use $\gamma = 4$ in this experiments. We adopt post-processing described in Section 3.5 separately to the results of every class. The results are shown in Table 3.5. The asymmetric focal loss can get better overall segmentation results than cross entropy or its symmetric variant. More importantly.

it can get better segmentation results with higher sensitivity for most rare classes. Specifically, asymmetric focal loss can improve the average DSC of rare classes by 4.9%. We also notice that asymmetric focal loss would decrease the segmentation performance of esophagus which is taken as a rare class. It is because esophagus is small and thin, and post-processing would remove the correct segmentation regions by mistake but leave the false positive predictions. We think more advanced post-processing would help improve the segmentation in this case.

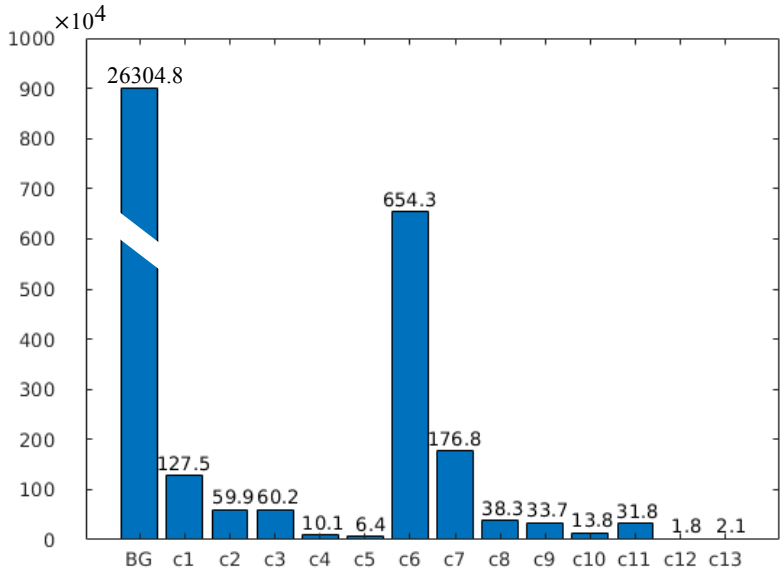


Figure 3.10: The class distribution of the abdominal organ dataset we use in this study. We summarize the total pixel number of different classes. We take class 4, 5, 8, 9, 10, 11, 12 and 13 as rare classes.

Table 3.5: Evaluation of abdominal organ segmentation with 25% of training data with symmetric (sy.) and asymmetric (asy.) focal loss. The rare classes are marked with $\boldsymbol{r}$. AVG is the average performance of all classes. AVG$_{\boldsymbol{r}}$ is the average performance of all rare classes. The best results are in **bold**.

	c1 (spleen)			c2 (right kidney)			c3 (left kidney)			c4 (gallbladder) $\boldsymbol{r}$			c5 (esophagus) $\boldsymbol{r}$		
	vanilla - CE	sy. focal loss	**asy. focal loss**	vanilla - CE	sy. focal loss	**asy. focal loss**	vanilla - CE	sy. focal loss	**asy. focal loss**	vanilla - CE	sy. focal loss	**asy. focal loss**	vanilla - CE	sy. focal loss	**asy. focal loss**
DSC	**85.3**	78.7	84.3	78.3	**84.4**	68.4	82.7	**85.0**	82.6	34.6	30.5	**53.7**	**55.7**	52.7	41.8
SEN	80.0	70.5	76.9	73.7	77.2	62.9	77.5	79.5	77.8	25.8	22.9	46.3	50.7	48.2	43.8
PRC	93.6	96.0	95.9	84.6	94.8	76.2	94.7	94.4	93.0	78.4	68.5	68.9	74.4	69.5	45.0
	c6 (liver)			c7 (stomach)			c8 (aorta) $\boldsymbol{r}$			c9 (vena cava) $\boldsymbol{r}$			c10 (vein) $\boldsymbol{r}$		
	vanilla - CE	sy. focal loss	**asy. focal loss**	vanilla - CE	sy. focal loss	**asy. focal loss**	vanilla - CE	sy. focal loss	**asy. focal loss**	vanilla - CE	sy. focal loss	**asy. focal loss**	vanilla - CE	sy. focal loss	**asy. focal loss**
DSC	87.4	88.4	**88.6**	39.7	39.7	**42.1**	82.7	80.3	**84.0**	65.1	66.4	**73.9**	42.0	26.3	**43.3**
SEN	84.1	85.2	84.0	28.8	28.5	31.0	76.6	73.1	82.9	56.8	60.3	76.9	28.4	16.5	31.0
PRC	92.3	92.8	94.2	91.3	84.1	86.3	91.6	91.5	86.3	86.1	79.3	72.7	91.5	82.1	80.0
	c11 (pancreas) $\boldsymbol{r}$			c12 (right adrenal) $\boldsymbol{r}$			c13 (left adrenal) $\boldsymbol{r}$			AVG			AVG$_{\boldsymbol{r}}$		
	vanilla - CE	sy. focal loss	**asy. focal loss**	vanilla - CE	sy. focal loss	**asy. focal loss**	vanilla - CE	sy. focal loss	**asy. focal loss**	vanilla - CE	sy. focal loss	**asy. focal loss**	vanilla - CE	sy. focal loss	**asy. focal loss**
DSC	17.2	24.3	**26.4**	54.3	32.8	**55.9**	34.8	28.9	**47.0**	58.5	55.3	**60.9**	48.3	42.8	**53.2**
SEN	11.1	17.3	18.3	45.3	24.3	50.3	27.2	22.2	41.6	51.2	48.1	55.7	40.2	35.6	48.9
PRC	56.6	52.0	61.7	74.5	60.8	69.4	61.4	52.7	63.6	82.4	78.3	76.4	76.8	69.6	68.5

3.6 Conclusion

We study overfitting of neural networks under class imbalance by inspecting network behavior. We observe that when processing unseen under-represented samples, the logit activations tend to shift towards the decision boundary and the sensitivity decreases. This phenomenon is confirmed across a variety of different tasks and two popular different network architectures. We derive simple yet effective asymmetric variants of existing loss functions and regularization techniques to prevent overfitting. We show that our proposed methods can substantially improve segmentation performance under class imbalance in terms of DSC and increased sensitivity, outperforming previous solutions.

We validate our proposed methods with extensive experiments of binary segmentation tasks and show promising improvements. When we extend our methods to multi-class segmentation such as kidney and kidney tumor segmentation, we find the proposed methods can improve the segmentation performance of specified rare class (kidney tumor), but could sometimes decrease the performance of other foreground class (kidney). We think it could be resolved by adopting a robust $\boldsymbol{r}$ for the network training. For example, we could consider to increase the rareness r_j for the kidney class. This could be a possible direction of future works.

In this study, we analyze the problem of class imbalance in the context of image segmentation, where the foreground class has disproportionate number of samples. The proposed loss functions

and regularization techniques are general for neural networks and could be extended to classification tasks. However, the performance needs to be demonstrated in the future. We believe more regularization methods can be derived to alleviate this problem by considering the biased network behavior. It would be particularly interesting to extend the variance of foreground samples by taking use of the geometric information and design data augmentation strategies with shape constraints to decrease the logit shifts.

We believe that the plotting logit distributions may be useful network inspection tool and help to gain a better understanding network behavior under different training scenarios. In future work, we will investigate if monitoring of intermediate activations may provide further insights for other challenging settings such as domain shift or self-supervised learning. In the next chapter, we would also utilize the logit plots to help us understand the over-segmentation problem caused by class imbalance.

Chapter 4

Underfitting under Class Imbalance

After investigating the reason of under-segmentation due to class imbalance in Chapter 3, here we look into the reasons of over-segmentation. The work in this chapter originates from the question of how the label space of the context information would affect the model training. Similar to the work in the previous chapter, here we aim to identify the effects of underfitting on the model behaviour. We investigate the representation learning of heterogeneous background classes. The work in this chapter is based on the publications:

1. **Z. Li**, K. Kamnitsas, C. Ouyang, C. Chen and B. Glocker. Context Label Learning: Improving Background Class Representations in Semantic Segmentation. IEEE Transactions on Medical Imaging, 2022, Provisionally Accepted.

Background samples provide key contextual information for segmenting regions of interest (ROIs). However, they always cover a diverse set of structures, causing difficulties for the segmentation model to learn good decision boundaries with high sensitivity and precision. The issue concerns the highly heterogeneous nature of the background class, resulting in multi-modal distributions. Empirically, we find that neural networks trained with heterogeneous background struggle to map the corresponding contextual samples to compact clusters in feature space. As a result, the distribution over background logit activations may shift across the decision boundary, leading to systematic over-segmentation across different datasets and tasks. In this study, we propose context label learning (CoLab) to improve the context representations by decomposing the background class into several subclasses. Specifically, we train an auxiliary network as a task generator, along with the primary segmentation model, to automatically generate context labels that positively affect the ROI segmentation accuracy. Extensive experiments are conducted on several challenging segmentation tasks and datasets. The results demonstrate that CoLab can guide the segmentation model to map the logits of background samples away from the decision boundary, resulting in significantly improved segmentation accuracy.

4.1 Introduction

Convolutional neural networks (CNNs) are the state-of-the-art approach for semantic image segmentation and other applications. Their large number of trainable parameters make them capable to fit to different kinds of tasks yielding high performance in terms segmentation accuracy [257]. Yet, in real world applications, CNNs seem sometimes unable to capture and generalize from complex, heterogeneous training sets when the amount of training data is limited [185]. Specifically, in the case of medical image segmentation, samples from the background class, which provide essential contextual information for regions of interest (ROIs), make up the majority of the training set while also containing diverse sets of structures with heterogeneous characteristics, making it hard for the segmentation model to learned accurate decision boundaries.

When trained with datasets with a highly heterogeneous background class, the segmentation model is prone to underfitting these contextual samples and failing to separate the ones which share characteristics similar to the ROI samples, which are usually one of the foreground samples. The model then produces false positives (FPs), yielding systematic over-segmentation. In this study, we argue that underfitting of the contextual information is a main cause of degraded segmentation performance by affecting precision (calculated as $\frac{TP}{TP+FP}$). We observe that better context representation where the background class is decomposed into several subclasses, for example, using additional anatomic labels, significantly improves the ROI segmentation accuracy. Context labels are labels of the background class which assign different labels to different background subregions. For medical imaging applications, context labels could be designed based on anatomy knowledge. We show some examples of human-defined context labels in Fig. 4.1(a). In the case of liver tumor segmentation, for example, it is beneficial to also have labels for the liver available in addition to the tumor class. Empirically, we find that when training with human-defined context labels, the segmentation model can yield better performance in terms of Dice similarity coefficient (DSC) and 95% Hausdorff distance (HD), as shown in Fig. 4.1(b). However, these human-defined context labels are not always available and difficult and time-consuming to obtain. In many applications, only the ROI labels, e.g., tumor class, are available. Here, we propose context label learning (CoLab), which automatically generates context labels to improve the learning of a context representation yielding better ROI segmentation accuracy. As demonstrated in Fig. 4.1(b), CoLab can bring similar improvements when compared with training with human-defined context labels, without the need for expert knowledge.

The contribution of this study can be summarized as follows: 1) With the observations of six datasets, we conclude that underfitting of the background class consistently degrades the segmentation performance by decreasing precision. 2) We find that better context representations with a decomposition of the background class can improve segmentation performance. 3) We propose CoLab, a flexible and generic method to automatically generate soft context labels. We

validate CoLab with extensive experiments and find consistent improvements where the segmentation accuracy is en par and sometimes better compared to the case where human annotated context labels are available.

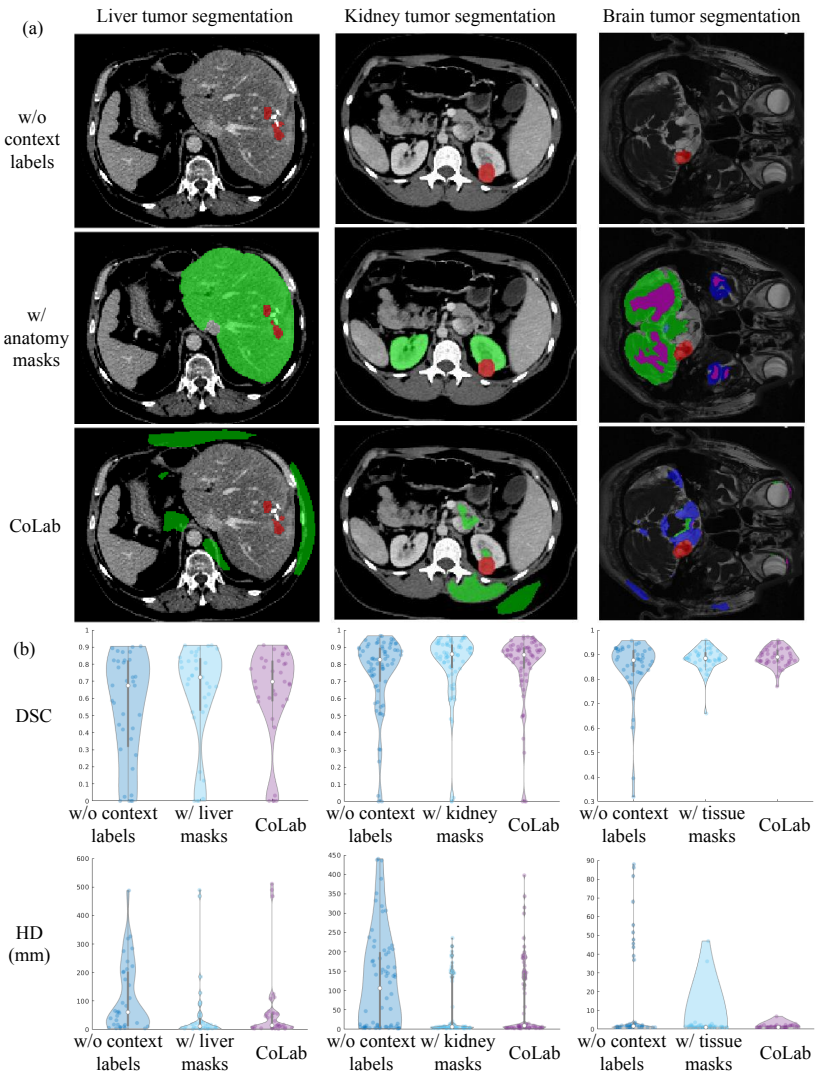


Figure 4.1: Effects of context labels on the segmentation accuracy. (a) Examples of ROI classes (in red color) along with human-defined context labels and context labels generated automatically by the proposed CoLab in background (in green, blue and other colors). Here, human-defined context labels (e.g., liver, kidney, brain tissue) provide additional anatomical information to benefit representation learning. In a similar manner, CoLab automatically finds and labels task-related tissues/structures in background to benefit corresponding ROI segmentation, see Fig. 4.9 for more details. (b) Self-supervised context labels generated by CoLab can significantly improve the segmentation of different structures with improved DSC and reduced HD, compared to the one trained w/o context labels. The performance gains are comparable and sometimes even higher than the ones using human-defined context labels.

4.2 Related Works

4.2.1 Class Imbalance

CoLab is related to the class imbalance problem, which we have introduced in Section 2.4.1, as background commonly constitutes the majority class in image segmentation. However, methods to combat class imbalance mostly focus explicitly on improving the performance of the minority categories [118, 164, 243]. Most approaches ignore the characteristics of majority background class as it is not contributing much to the common evaluation metrics of sensitivity, precision, and DSC of the ROI classes. CoLab focuses specifically on the representation of the background class and is complementary to methods tackling class imbalance such as loss reweighting strategies [29].

Previous studies adopt coarse-to-fine strategies to reduce FP in segmentation tasks with class imbalance [210, 231, 270]. However, ROI samples missed in the coarse stage cannot be recovered with later stages. In contrast, CoLab is trained in an end-to-end manner and can reduce varied kinds of FPs.

4.2.2 Multi-task Learning

CoLab, which is formulated as multi-label classification, can be seen as a form of multi-task learning (MTL). Current MTL methods train the model with different predefined tasks together with the main task using a shared feature representation [23, 151, 196]. Previous works also attempted to incorporate spatial prior [78, 259] or task prior [258, 269] into model training with some predefined auxiliary tasks and optimization functions. In contrast, CoLab reformulates the main task by decomposing the background class with context labels and automatically generate the auxiliary task in a self-supervised manner. We argue that CoLab can have a direct impact on the main task by extending the label space.

The main methodology of the CoLab strategy is inspired by some recently proposed methods which aim to generate the weights for pre-defined auxiliary tasks or labels through a similar meta-learning framework [160, 182]. In this study, CoLab is specifically designed for semantic segmentation with heterogeneous background classes which is a common scenario in medical imaging.

4.3 Analysis

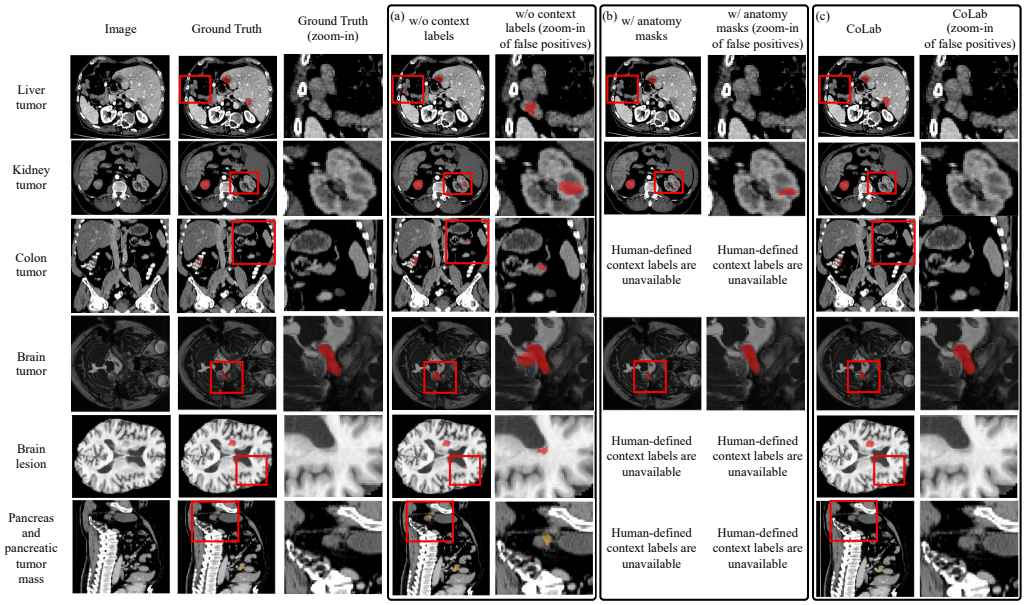


Figure 4.2: Visualization of different datasets and segmentation results on test data when training w/o and w/ different types of context labels. (a) Models trained without any context labels are prone to over-segmentation of the ROI (red/red + orange) with different kinds of FP (marked with red boxes). For reference, we also zoom-in the corresponding regions of the ground truth label map. (b) Empirically, we observe that adding human-defined context labels such as anatomy masks for supervision helps reduce FP and therefore obtains higher overall segmentation accuracy. (c) The proposed CoLab achieves a similar benefit without requiring additional human annotation by producing context labels automatically.

4.3.1 Preliminaries

We consider CNNs for multi-class segmentation with a total of c classes. Given a training dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$ with N samples, where $\boldsymbol{y}_i \in \mathbb{R}^{\mathrm{c}}$ is the one-hot encoded label for the central pixel in the image sample $\boldsymbol{x}_i \in \mathbb{R}^{\mathrm{d}}$, such that $\mathbf{1} \cdot \boldsymbol{y}_i = 1 \; \forall \; i$. A segmentation model $f_\phi(\cdot)$ learns class representations of the input sample $\boldsymbol{x}_i$, noted as $\boldsymbol{z}_i = f_\phi(\boldsymbol{x}_i) \in \mathbb{R}^{\mathrm{c}}$. We obtain the predicted probability $\boldsymbol{p}_i$ that the real class of $\boldsymbol{x}_i$ is j via a softmax function with $p_{ij} = \mathrm{e}^{z_{ij}} / \sum_{j=1}^{c} \mathrm{e}^{z_{ij}}$. Typically the model is optimized by minimizing the empirical risk $R_{\mathcal{L}_{seg}}(f_\phi) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{seg}(f_\phi(\boldsymbol{x}_i), \boldsymbol{y}_i)$ computed on the training set. The segmentation loss $\mathcal{L}_{seg}$ can be defined as the sum of losses $\mathcal{L}$ over c classes:

$$\mathcal{L}_{seg}(f_\phi(\boldsymbol{x}_i), \boldsymbol{y}_i) = \sum_{j=1}^{c} \mathcal{L}(p_{ij}, y_{ij}) = \underbrace{\sum_{j=1}^{c-1} \mathcal{L}(p_{ij}, y_{ij})}_{\text{ROI classes}} + \underbrace{\mathcal{L}(p_{ic}, y_{ic})}_{\text{background class}}, \tag{4.1}$$

where $\mathcal{L}$ is a criterion for a specific class, such as cross entropy (CE) or soft DSC [175]. Here, we further decompose $\mathcal{L}_{seg}$ into two terms, including an ROI loss (computed on $c-1$ foreground classes) and a background loss (computed on the background class, only).

We aim to improve segmentation performance by augmenting the background class with auxiliary context classes. Specifically, we propose to utilize context labels assigned to different and decomposed background regions. In order to divide the background class into $t > 1$ classes, we create another model $\tilde{f}_\theta(\cdot)$ with $\tilde{\boldsymbol{z}}_i = \tilde{f}_\theta(\boldsymbol{x}_i) \in \mathbb{R}^{c+t-1}$ and predicted probability $\tilde{\boldsymbol{p}}_i$. We also require an additional one-hot label $\tilde{\boldsymbol{y}}_i \in \mathbb{R}^{c+t-1}$, where we require $\tilde{y}_{ij} = y_{ij}\ \forall\, i, j \in \{1, \cdots, c-1\}$ and $\sum_{j=c}^{c+t-1} \tilde{y}_{ij} = y_{ic}\ \forall\, i$. With this notion, the segmentation loss can be written as:

$$\mathcal{L}_{seg}(\tilde{f}_\theta(\boldsymbol{x}_i), \tilde{\boldsymbol{y}}_i) = \sum_{j=1}^{c+t-1} \mathcal{L}(\tilde{p}_{ij}, \tilde{y}_{ij}) = \underbrace{\sum_{j=1}^{c-1} \mathcal{L}(\tilde{p}_{ij}, \tilde{y}_{ij})}_{\text{ROI classes}} + \underbrace{\sum_{j=c}^{c+t-1} \mathcal{L}(\tilde{p}_{ij}, \tilde{y}_{ij})}_{\text{background classes}}, \tag{4.2}$$

In the following method section, we always consider a simplified but common case where we only have one ROI class ($c = 2$) for simplicity. In other words, we only consider binary ROI segmentation with $f_\phi(\boldsymbol{x}_i) \in \mathbb{R}^2$, although the method can be naturally extended to multi-class ROI segmentation. With this assumption, Eq. 4.2 can be simplified as:

$$\mathcal{L}_{seg}(\tilde{f}_\theta(\boldsymbol{x}_i), \tilde{\boldsymbol{y}}_i) = \underbrace{\mathcal{L}(\tilde{p}_{i1}, \tilde{y}_{i1})}_{\text{ROI class}} + \underbrace{\sum_{j=2}^{t+1} \mathcal{L}(\tilde{p}_{ij}, \tilde{y}_{ij})}_{\text{background classes}} \tag{4.3}$$

4.3.2 False Positives on Background Class

To better understand the effect of the background class on the model learning, we train multiple CNNs on segmentation datasets which contain heterogeneous background. We conduct experiments on challenging tasks including liver tumor segmentation in computerised tomography (CT) images [18], kidney tumor segmentation in CT images [96], colon tumor segmentation in CT images [219], vestibular schwannoma (VS) segmentation in T2-weighted magnetic resonance (MR) images [212], brain stroke lesion segmentation in T1-weighted MR images [152] and pancreatic

tumor segmentation in CT images [219]. We adopt a three-dimensional (3D) U-Net [106] as the segmentation model for all the experiments, which has been demonstrated to yield competitive results across different medical image segmentation tasks. The detailed data and network configurations are summarized in Section 4.5.

Visualizations of the segmentation results are shown in Fig. 4.2(a). When trained with binary segmentation tasks and without context labels, the models are prone to over-segment the ROIs with many FPs. Specifically, the model trained on the binary liver tumor task predicts other organs outside liver as liver tumor; the model trained on the binary kidney tumor task predicts parts of the healthy kidney regions as kidney tumor; while the model trained on the binary brain tumor task predicts the surrounding brain tissue as brain tumor; the model trained on the binary brain lesion task predicts unrelated healthy brain regions as brain lesion.

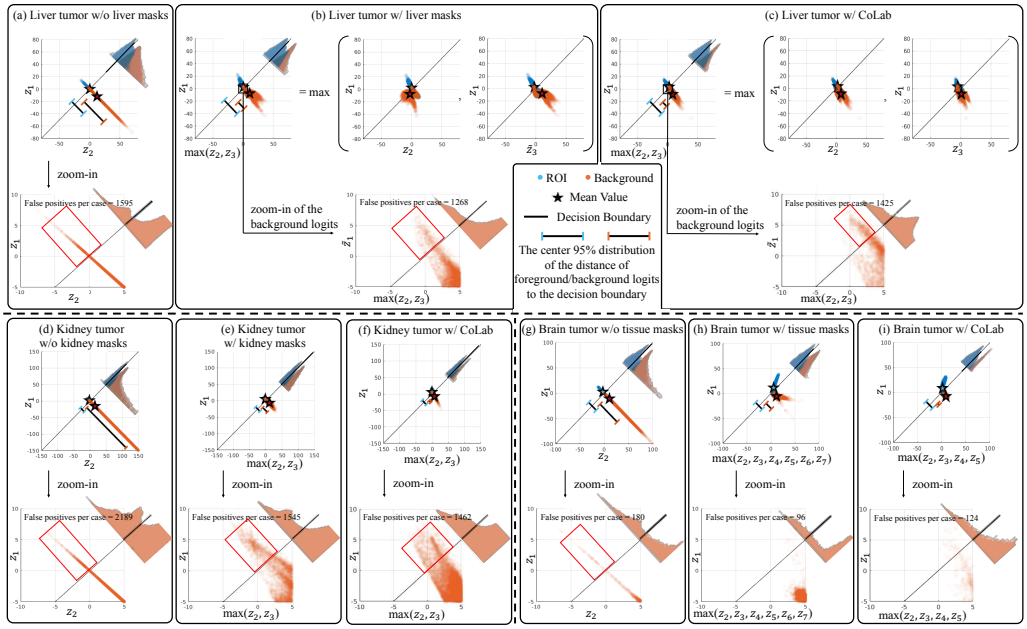


Figure 4.3: The distribution of activations (logits) after the classification layer using ROI (blue) or background samples (orange) of test data as network input. Here, z_1 denotes logits for ROI classes, while z_i $(i > 1)$ denotes logits for background classes. We also report the number of pixel-level predictions that are FP. Without context labels (a, d, g), CNNs fail to map the background samples to compact clusters. This is because CNNs are forced to learn generic filters for the background samples, yielding similar activations regardless of different background structures with diverse patterns/appearance. Consequently, the logit distribution of background samples would spread across the decision boundary, leading to increased FP. Extending the label space with anatomy masks (b, e, h) or CoLab (c, f, i) can alleviate this issue.

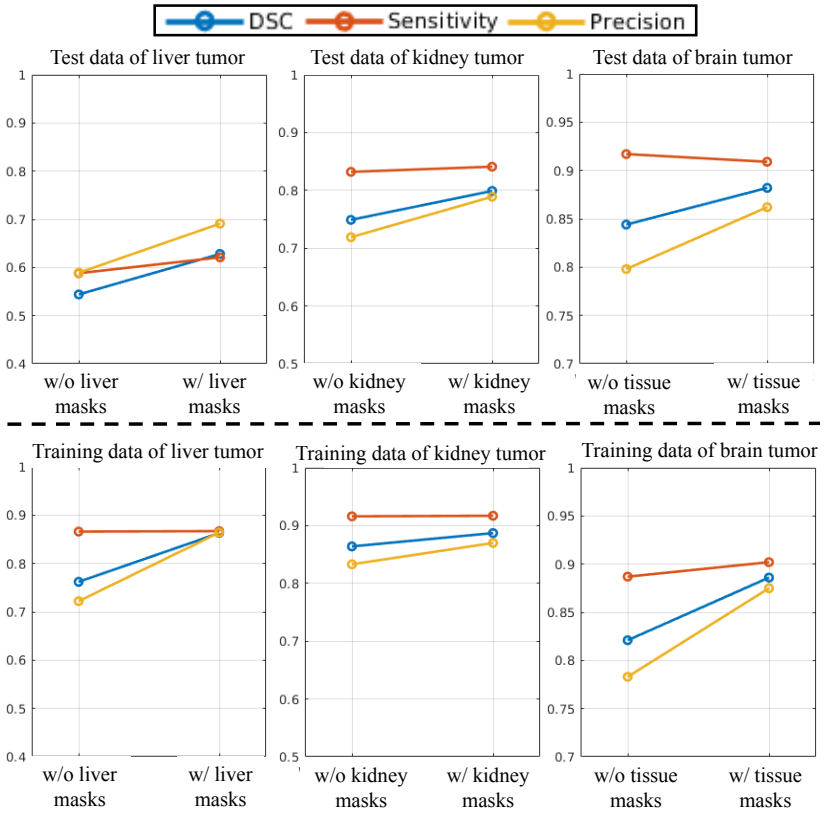


Figure 4.4: Effects of context labels on the learning of neural networks. Performance on liver tumor, kidney tumor and brain tumor segmentation on training data (lower part) and testing data (upper part) when networks were trained w/o and w/ context labels. Extending the label space with anatomy masks can help the model improve the performance by increasing the precision, while the sensitivity is largely retained.

4.3.3 Underfitting of Background Samples

To study the model behavior when trained with heterogeneous background classes, we can monitor the logit distribution of samples from ROI and background classes for the test data. Our observations for liver tumor, kidney tumor, and brain tumor segmentation are summarized in Fig. 4.3(a, d, g). We find that the CNN models map the ROI samples to a compact cluster in the logit space while background samples form a more dispersed distribution. This indicates that the model cannot easily map all background samples to a single cluster representing the background class. Although the model seems to separate ROI from background samples in the feature space, it builds complex background representations and unable to capture an accurate decision boundary between ROI and immediate context. A possible reason is that the CNN uses most of its capacity to extracted the common features among background regions with different characteristics. Unfortunately, these shared features are not very discriminative. Specifically, we observe that the logit distribution of background samples overlaps with the learned decision boundary. This is the reason why

the model predicts many FP leading to over-segmentation of ROI structures when the background class is heterogeneous. We hypothesise that the width of the distribution serves as an indicator of the heterogeneity of a specific class and is a sign of the difficulty during learning. As a result of sample heterogeneity, a CNN may struggle to reduce intra-class variation of the background class and underfit the background samples, failing to recognize the background samples that share similar characteristics with ROI samples.

It should be noted that we do not observe significant difference between the logit distribution of training background samples and test background samples, as also shown in previous studies [146, 147], indicating that the logit shift of background samples is indeed due to underfitting instead of overfitting. Specifically, we show the network behaviour with the training data in Fig. 4.5. The behaviour is similar to the ones with test data. We observe little performance gap between the training and test data. This indicates that the reason why the model cannot recognize the background sample well is that it underfits the training data. In consequence, the model is likely to make false positives even for the training data. We find anatomy masks and CoLab both can alleviate this issue.

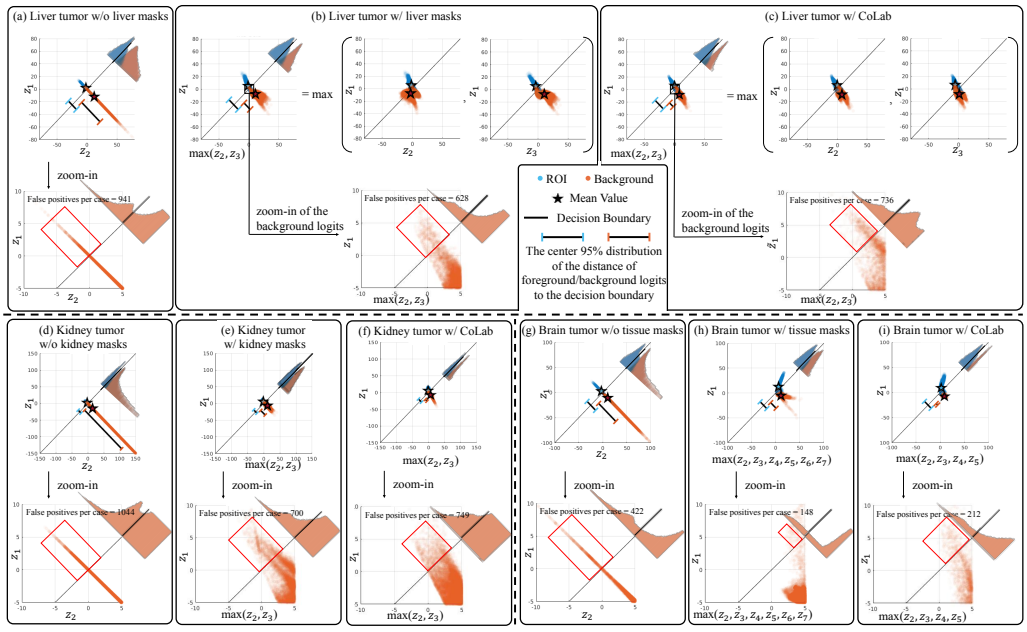


Figure 4.5: The distribution of activations (logits) after the classification layer using ROI (blue) or background samples (orange) of training data as network input. Activations of the classification layer (logit z_1 for ROI, other logits for background) when processing the ROI (blue) or background sample (orange) of training data. It seems difficult for a CNN to learn good representations over heterogeneous background samples due to the varying characteristics leading to diverse feature embeddings. Extending the label space with anatomy masks or CoLab can help CNN reduce FPs and fit the training data better.

4.3.4 The Effect of Context Labels

The availability of context labels greatly helps with the ROI segmentation, which provide additional signals to CNN to fit the training data of heterogeneous background samples. We confirm this empirically by including human-defined context labels in the model training. Specifically, we adopt liver masks ($t = 2$) for liver tumor segmentation, kidney masks ($t = 2$) for kidney tumor segmentation, brain tissue masks including ventricles, deep grey matter, cortical grey matter, white matter and other tissues ($t = 6$) for brain tumor segmentation. The kidney and liver masks are manually annotated, while the brain tissue masks are generated automatically using the paired T1-weighted MR images and a multi-atlas label propagation with expectation–maximisation based refinement (MALP-EM) [136]. In order to make a fair comparison, we make sure that all the experiments share the same training schedule except the label space. We sample the training patches only considering ROIs. As the image size is quite large, here we train the 3D U-Net with patches which are sampled from the training data. Specifically, we make sure 50% training patches to contain ROIs and sample the other half of the training patches uniformly.

The observations on test and training set are summarized in Fig. 4.4. We find that the models yield overall better performance when trained with anatomy masks, as indicated by improved DSC (defined as $\text{DSC} = 2\frac{\text{SEN}\cdot\text{PRC}}{\text{SEN}+\text{PRC}}$). Furthermore, we observe that the model trained with context labels yields higher precision while preserving similar sensitivity. The observation is consistent with the visualization results in Fig. 4.2(b), where we find the models trained with human-defined context labels reduce FP.

Similarly, we visualize the corresponding network behaviour in Fig. 4.3(b, e, h). As we only consider the segmentation performance of the ROI (class 1), we visualize the logits in the plane of (z_1, $\max(z_2, ..., z_{t+1})$). We observe that the model trained with anatomy masks maps the background samples to a narrower distribution and reduces the background logits shift across the decision boundary. This indicates that the models fit the training data better with the help of context labels. Instead of building generic filters for all the background samples, the CNNs can dedicate specific filters to model a more homogeneous subparts of the background samples that share common characteristics. The models are faced with a simplified segmentation task with homogeneous background subclasses yielding better overall performance with the same model capacity.

Although anatomy masks are found to be effective context labels, they are not always available in real-world applications. Specifically, acquiring manually annotated context labels is time-consuming and would require significant efforts from human experts to generate annotations at large scale. Therefore, we propose CoLab which can automatically discover specific soft context labels using a meta-learning strategy. CoLab benefits the segmentation model training by making it to fit the background samples better, achieving comparable or even better performance when compared with models trained on manually defined anatomy masks.

4.4 Method

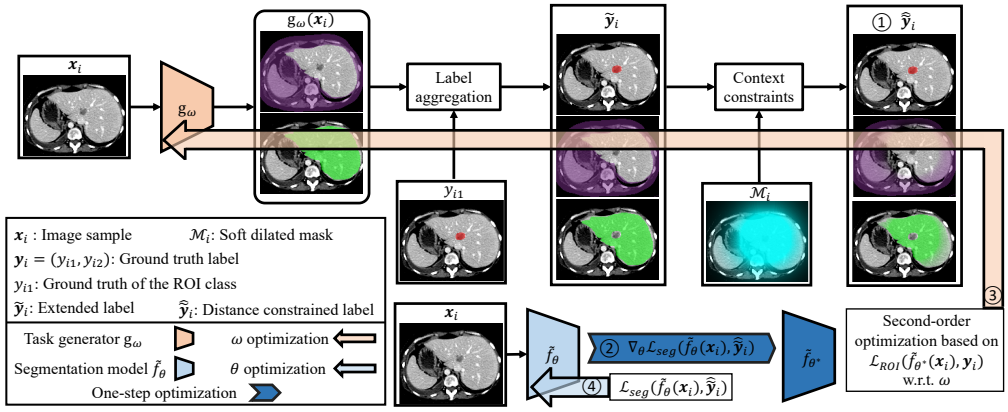


Figure 4.6: The illustration of the training process of CoLab $g_\omega(\cdot)$ is optimized to generate context labels such that it can help reduce the training loss of the ROI in the most effective way, see Section 4.4 for more details.

Now we consider CNNs for binary semantic segmentation. Typically, we are given a segmentation model $f_\phi(\cdot)$ that maps the input image $\boldsymbol{x}_i$ to the label space $f_\phi(\boldsymbol{x}_i) \in \mathbb{R}^2$. In order to fit the label space with context labels, we first extend the classification layer of $f_\phi(\cdot)$ with additional $t-1$ output neurons and obtain $\tilde{f}_\theta(\cdot)$ which map $\boldsymbol{x}_i$ to $\tilde{f}_\theta(\boldsymbol{x}_i) \in \mathbb{R}^{t+1}$. We employ another model $g_\omega(\cdot)$ as the task generator parameterized by ω to produce context labels, with the network output $\boldsymbol{o}_i = g_\omega(\boldsymbol{x}_i) \in \mathbb{R}^t$. We have no requirements for the backbone of $g_\omega(\cdot)$ and empirically keep it the same with that of $f_\phi(\cdot)$.

We illustrate the training process of the proposed CoLab in Fig. 4.6. At the beginning of each iteration, we obtain the context labels termed as distance constrained label $\hat{\tilde{\boldsymbol{y}}}_i$ by taking use of $\boldsymbol{o}_i$ and the ground truth $\boldsymbol{y}$. This process is marked with ① and will be illustrated in Section 4.4.1 and 4.4.2. Then, we calculate a new θ^* with one step of gradient descent with ② to access the impact of $\hat{\tilde{\boldsymbol{y}}}_i$ on model training of ROI segmentation. Next, we optimize ω based on the second-order derivatives through a meta learning scheme with ③, which would be demonstrated in Section 4.4.3. Finally, we optimize the segmentation model $\tilde{f}_\theta(\cdot)$ based on the updated context labels with ④.

4.4.1 Label Aggregation

We calculate the context probability $(q_{ij})_{j=1}^t$ based on $\boldsymbol{o}_i$ via the softmax function as:

$$q_{ij} = \frac{\mathrm{e}^{o_{ij}}}{\sum_{j=1}^{t} \mathrm{e}^{o_{ij}}}. \tag{4.4}$$

The extended label $\tilde{\boldsymbol{y}}_i \in \mathbb{R}^{t+1}$ is calculated by aggregating the original label $\boldsymbol{y}_i = (y_{i1}, y_{i2})$ and

the context probability $\boldsymbol{q}_i$ with:

$$\tilde{y}_{ij} = \begin{cases} y_{i1} & \text{if} \quad j = 1, \\ q_{ij} & \text{if} \quad j > 1 \quad \text{and} \quad y_{i1} = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{4.5}$$

The label aggregation process is also illustrated in Fig. 4.7. By doing so, we can decompose the background class y_{i2} into t subclasses while ensure that the $\tilde{\boldsymbol{y}}_i$ contains sufficient information about the ROI segmentation. For multi-class segmentation with totally c classes, $\tilde{\boldsymbol{y}}_i$ can be calculated as:

$$\tilde{y}_{ij} = \begin{cases} y_{ij} & \text{if} \quad j < c, \\ q_{ij} & \text{if} \quad j \geq c \quad \text{and} \quad y_{ij} = 0 \quad \forall j < c, \\ 0 & \text{otherwise.} \end{cases} \tag{4.6}$$

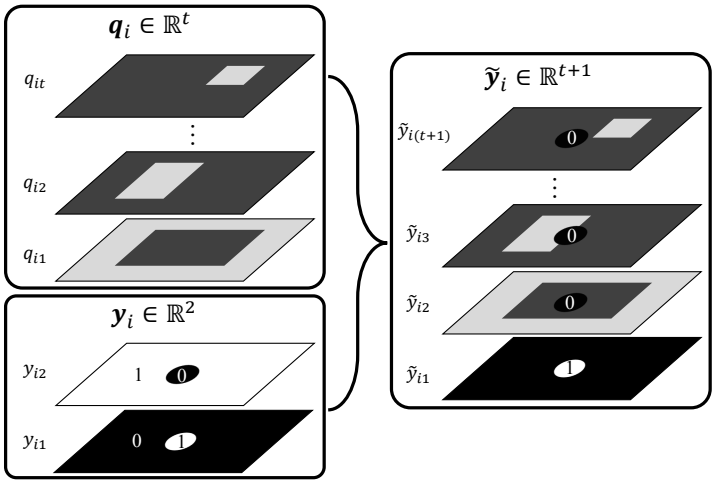


Figure 4.7: The illustration of the label aggregation process. We generate the extended label $\tilde{\boldsymbol{y}}_i$ by aggregating the original label $\boldsymbol{y}_i$ with the context probability $\boldsymbol{q}_i$.

4.4.2 Context Constraints

Compared with background samples which are further away, the ones closer to the ROIs share similar characteristics with ROI samples and are more likely to be misclassified. In order to make the segmentation model focus more on those hard background samples that are close to ROIs , we make an assumption that all samples which are distant from the ROI need less attention and should be safely assigned the same background label. Specifically, we create a hard label $\boldsymbol{b}_i \in \mathbb{R}^{t+1}$ to represent the the background samples that are far from the ROI:

$$b_{ij} = \begin{cases} 1 & \text{if} \quad j = 2, \\ 0 & \text{otherwise.} \end{cases} \tag{4.7}$$

By utilizing the ROI label $\boldsymbol{y}_i$, we calculate the corresponding distance map d_i which is the Euclidean distance of a pixel to the closest boundary point of the ROI of any class. We set d_i to be zero for pixels inside the ROI.

We then calculate a soft dilated mask $\mathcal{M}_i$ based on d_i:

$$\mathcal{M}_i = \begin{cases} 1 & \text{if} \quad d_i < m, \\ \mathrm{e}^{\frac{-d_i+m}{\tau}} & \text{otherwise,} \end{cases} \tag{4.8}$$

where m is the margin controlling the model's focus on the pixels which are neighbouring to ROI and τ is the temperature to control the probabilities of the dilated regions. Empirically, m is set as 30 units and τ is set as 20 for all our experiments. $\mathcal{M}_i$ would be 1 for pixels around ROIs and decrease close to 0 for pixels far from ROIs . We visualize an example of $\mathcal{M}_i$ with liver tumor in Fig. 4.8. The distance constrained label then is calculated as:

$$\hat{\tilde{\boldsymbol{y}}}_i = \mathcal{M}_i \tilde{\boldsymbol{y}}_i + (1 - \mathcal{M}_i)\boldsymbol{b}_i. \tag{4.9}$$

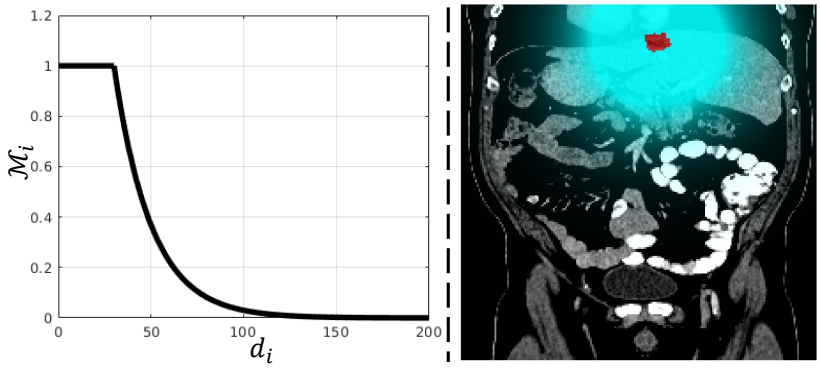


Figure 4.8: The illustration of context constraints. The task generator focuses on the context which is near to the ROI. To do so, we train the segmentation model to classify the pixels which are outside the soft dilated ROI as the same class (pixels marked without color in the right figure). (Left part) The value of a soft dilated mask $\mathcal{M}_i$ with $m = 30$ and $\tau = 20$. (Right part) The visualization of the dilated mask $\mathcal{M}_i$ (blue) with a liver tumor (red) of a training case.

In this way, only the regions neighbouring to the tumor are considered to be classified as the contextual background class. Specifically, both $\tilde{f}_\theta(\cdot)$ and $g_\omega(\cdot)$ would be trained to focus on the regions which are close to ROIs .

4.4.3 Task Generator Optimization with Meta-Gradients

We formulate the optimization of CoLab as a bi-level problem:

$$\min_{\omega} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{ROI}(\tilde{f}_{\theta^*}(\boldsymbol{x}_i), \boldsymbol{y}_i) \tag{4.10}$$

$$s.t. \quad \theta^* = \underset{\theta}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{seg}(\tilde{f}_{\theta}(\boldsymbol{x}_i), \hat{\tilde{\boldsymbol{y}}}_i)), \tag{4.11}$$

where $\mathcal{L}_{ROI}(\tilde{f}_{\theta}(\boldsymbol{x}_i), \boldsymbol{y}_i) = \mathcal{L}_c(\tilde{p}_{i1}, y_{i1})$ is computed on the ROI class. Different from $\mathcal{L}$, we choose $\mathcal{L}_c$ to be a criteria that can be calculated in the one-versus-all manner, such as binary cross entropy (BCE) and soft DSC loss, to represent the binary segmentation performance of the ROI class with more than two output logits.

We train the model using a batch of training samples with batch size n. For simplicity, we shorten $\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_{ROI}(\tilde{f}_{\theta^*}(\boldsymbol{x}_i), \boldsymbol{y}_i)$ as $\mathcal{L}_{ROI}(\theta^*)$ and $\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_{seg}(\tilde{f}_{\theta}(\boldsymbol{x}_i), \hat{\tilde{\boldsymbol{y}}}_i)$ as $\mathcal{L}_{seg}(\theta, \omega)$ in the following descriptions. The bi-level optimization problem defined in Eq. 4.10 and 4.11 can be solved with gradient descent [74, 192]. Specifically, the derivative of $\mathcal{L}_{ROI}(\theta^*)$ w.r.t. ω can be calculated by applying the chain rule:

$$\nabla_{\omega} \mathcal{L}_{ROI}(\theta^*) = (\frac{\partial \theta^*}{\partial \omega})^{\intercal} \nabla_{\theta} \mathcal{L}_{ROI}(\theta^*), \tag{4.12}$$

where $\nabla_{\omega} = (\frac{\partial}{\partial \omega})^{\intercal}$ and $\nabla_{\theta} = (\frac{\partial}{\partial \theta})^{\intercal}$. One can compute $\frac{\partial \theta^*}{\partial \omega}$ based on implicit function theorem [15]. However, the derived result would contain a Hessian which is computational expensive and not always possible to access for deep neural networks. Among the many heuristics for the gradient approximation [74, 157, 192], we follow the solution described in [73, 157] to approximate θ^* by a single optimization step. Specifically, we sample a batch of training data and approximate the optimal inner variable θ^* with a step of gradient decent:

$$\theta^* \approx \theta - \alpha \nabla_{\theta} \mathcal{L}_{seg}(\theta, \omega), \tag{4.13}$$

where α is step size, which is kept the same with the learning rate of θ. We differentiate this equation w.r.t. ω from both sides yielding:

$$\frac{\partial \theta^*}{\partial \omega} = -\alpha \nabla^2_{\theta,\omega} \mathcal{L}_{seg}(\theta^*, \omega), \tag{4.14}$$

where $\nabla^2_{\theta,\omega} = \frac{\partial \nabla_{\theta}}{\partial \omega}$. By substituting Eq. 4.14 into Eq. 4.12, we can obtain the gradient on ω and

update it with:

$$\begin{aligned}\omega^{t+1} &= \omega^t - \beta\nabla_{\omega^t}\mathcal{L}_{ROI}(\theta^*) \\ &= \omega^t + \alpha\beta\nabla^2_{\omega^t,\theta}\mathcal{L}_{seg}(\theta^*,\omega^t)\nabla_\theta\mathcal{L}_{ROI}(\theta^*),\end{aligned} \tag{4.15}$$

where β is the learning rate to update ω. In this way, the task generator g_ω is explicitly trained to produce effective context labels with the second-order gradients. Although Eq. 4.15 contains an expensive vector-matrix product, it is feasible to calculate with prevailing machine learning frameworks such as PyTorch [191]. We find CoLab can be efficient as it costs as little as 30% additional training time. The full algorithm is summarized in Algorithm 1.

Algorithm 1 CoLab

Require:
$\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$: training data; t: the number of context classes, $f_\phi(\cdot)$: the segmentation model which produces $f_\phi(\boldsymbol{x}_i) \in \mathbb{R}^2$; $g_\omega(\cdot)$: the task generator which produce $g_\omega(\boldsymbol{x}_i) \in \mathbb{R}^t$. α, β: learning rates to update θ and ω.

Extend the classification layer of $f_\phi(\cdot)$ to get $\tilde{f}_\theta(\cdot)$ which produce $\tilde{f}_\theta(\boldsymbol{x}_i) \in \mathbb{R}^{t+1}$.
for each iteration **do**
  Sample a batch of data $\mathcal{B} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$ from $\mathcal{D}$.
  for a number of steps **do** ▷ *Note: One step is sufficient in our experiments.*
    Generate the context probability $\boldsymbol{q}_i$ based on the output of task generator $g_\omega(\boldsymbol{x}_i)$ via Eq. 4.4.
    Generate the extended label $\tilde{\boldsymbol{y}}_i$ by aggregating the context probability $\boldsymbol{q}_i$ with $\boldsymbol{y}_i$ via Eq. 4.5.
    Calculate a loss mask $\mathcal{M}_i$ via Eq. 4.8 and generate the distance constrained label $\hat{\tilde{\boldsymbol{y}}}_i$ via Eq. 4.9.
    Calculate a new θ^* with the gradient descent optimization algorithm via Eq. 4.13.
    Optimize ω based on meta-gradients via Eq. 4.15. ▷ *Training the task generator.*
  end for
  Update θ with gradient decent based on updated ω. ▷ *Training the segmentation model.*
end for

4.4.4 Implementation Details

4.4.4.1 Derivatives calculation based on implicit function theorem

One can obtain $\frac{\partial\theta^*}{\partial\omega}$ based on implicit function theorem [15]. In order to compute $\frac{\partial\theta^*}{\partial\omega}$, one can assume that $\nabla_\theta\mathcal{L}_{seg}(\theta^*,\omega)$ is continuously differentiable around $\mathbf{0}$ w.r.t. ω following [255], and calculate the total derivative of $\nabla_\theta\mathcal{L}_{seg}(\theta^*,\omega) = \mathbf{0}$ w.r.t. ω from both sides:

$$\frac{\partial\nabla_\theta\mathcal{L}_{seg}(\theta^*,\omega)}{\partial\theta}\frac{\partial\theta^*}{\partial\omega} + \frac{\partial\nabla_\theta\mathcal{L}_{seg}(\theta^*,\omega)}{\partial\omega} = \mathbf{0}. \tag{4.16}$$

With the assumption that $\nabla^2_\theta\mathcal{L}_{seg}(\theta^*,\omega)$ is invertible, one can yield:

$$\frac{\partial \theta^*}{\partial \omega} = -(\nabla^2_{\theta}\mathcal{L}_{seg}(\theta^*, \omega))^{-1}\nabla^2_{\theta,\omega}\mathcal{L}_{seg}(\theta^*, \omega), \tag{4.17}$$

However, the Hassian $\nabla^2_{\theta}\mathcal{L}_{seg}(\theta^*, \omega)$ is expensive to compute. Therefore, in this study we follow some heuristics such as [73] to approximate $\frac{\partial \theta^*}{\partial \omega}$.

4.4.4.2 Meta-gradient calculation

In our study, we further simplify the calculation of Eq. 4.15 with the finite difference approximation following [157]. We calculate network parameters $\theta^{\pm} = \theta \pm \epsilon\nabla_{\theta}\mathcal{L}_{ROI}(\theta^*)$ given some small scalars $\epsilon = 0.01/\left\|\nabla_{\theta}\mathcal{L}_{ROI}(\theta^*)\right\|_2$. With this notion, the second-order gradient can be written as:

$$\begin{aligned} &\nabla^2_{\omega^t,\theta}\mathcal{L}_{seg}(\theta^*, \omega^t) \approx \frac{\nabla_{\omega^t}\mathcal{L}_{seg}(\theta^+, \omega^t) - \nabla_{\omega^t}\mathcal{L}_{seg}(\theta^-, \omega^t)}{2\epsilon\nabla_{\theta}\mathcal{L}_{ROI}(\theta^*)} \\ \Leftrightarrow \quad &\nabla^2_{\omega^t,\theta}\mathcal{L}_{seg}(\theta^*, \omega^t)\nabla_{\theta}\mathcal{L}_{ROI}(\theta^*) \approx \frac{\nabla_{\omega^t}\mathcal{L}_{seg}(\theta^+, \omega^t) - \nabla_{\omega^t}\mathcal{L}_{seg}(\theta^-, \omega^t)}{2\epsilon}. \end{aligned} \tag{4.18}$$

In this way, we reduce the calculation complexity from $O(|\omega||\theta|)$ to $O(|\omega| + |\theta|)$ and can approximate Eq. 4.15 with two forward passing of $\tilde{f}_{\theta^*}$.

4.4.4.3 Update of task generator

We find the segmentation model would not learn well if the context labels changes too frequently. Therefore, we only update the task generator once several epochs. Specifically, we only update ω following step 8 in Algorithm 1 once 10 epochs. Another benefit of updating ω once several epochs is that we do not need much more time than the vanilla training process. In practice, we find CoLab just adds extra 20% to 30% of the vanilla training time.

4.5 Experiments

4.5.1 Experimental Setup

4.5.1.1 Network configurations

We use a state-of-the-art (SOTA) 3D U-Net [106] as the network backbone for both the segmentation model $f_\phi/\tilde{f}_\theta$ and the task generator g_ω. We normalize all datasets with the built-in pipeline following [106]. Specifically, we adopt case-wise Z-score normalization within brain masks for MR images, while we employ dataset-wise Z-score normalization based on ROI samples for CT images after clipping the Hounsfield units (HU) from 0.5% to 99.5%. We utilize the default data augmentation policies for all experiments. We adopt a combination of CE and sample-wise soft DSC loss with equal weight for $\mathcal{L}$ while using BCE for $\mathcal{L}_c$. We use a batch size of 2 and patch size of $80\times80\times80$. We train the networks for 1000 epochs for brain tumor and lesion segmentation and 2000 epochs for liver, kidney, colon and pancreas tumor segmentation as we observed that the network needed more iterations to converge latter tasks. All reported results are the average of two runs with different random seeds.

4.5.1.2 Liver tumor segmentation

We evaluate CoLab for liver tumor segmentation with the training dataset from the liver tumor segmentation (LiTS) challenge which contains 131 CT images. We exclude the samples which do not contain any liver tumor leaving 118 cases. We resample all CT images to a common voxel spacing of $1.9\times1.9\times2.5$ mm following [106]. We train models with 83 cases and test on 35 cases.

4.5.1.3 Kidney tumor segmentation

We further conduct experiments using the training dataset of the kidney tumor segmentation (KiTS) challenge which contains 210 CT images. We resample all CT images to a common voxel spacing of $1.6\times1.6\times3.2$ mm following [106]. We tested on 70 cases and used the other 140 cases as the training data.

4.5.1.4 Colon tumor segmentation

We evaluate CoLab for the case of colon tumor segmentation from CT images. We collect 126 colon cancer CT images from the training dataset of the Medical Segmentation Decathlon challenge [219]. We resample all CT images to the voxel spacing of $1.6\times1.6\times3.1$ mm following [106]. We train models with 88 cases and test on the other 38 cases.

4.5.1.5 Brain tumor segmentation

We also conduct experiments for brain tumor segmentation using the VS dataset [212] which contains 243 paired T1-weighted and T2-weighted MR images. We only used T2-weighted MR images for evaluating CoLab but used T1-weighted MR images to generate the brain structure masks with MALP-EM. We did not use brain masks for the histogram normalization of this task because VS can appear at the brain boundary and may be excluded when using common brain extraction algorithms. All MR images have the same isotropic voxel spacing of 1.0 mm^3. We train the models with 176 cases and test on 46 cases following [212].

4.5.1.6 Brain stroke lesion segmentation

Additionally, we evaluate CoLab with brain stroke lesion segmentation using the anatomical tracings of lesions after stroke (ATLAS) dataset [152] which contains 220 T1-weighted MR images. The MR images all have the same voxel spacing of 1.0 mm^3. We randomly selected 145 cases as training data and left the rest 75 cases for testing.

4.5.1.7 Pancreas and pancreatic tumor mass segmentation

We further evaluate CoLab in the setting of multi-class segmentation. Specifically, we train the segmentation model to segment three classes including pancreas, pancreatic tumor mass and background. We aim to find the context labels which can benefit the segmentation of both foreground classes. We collect 281 CT images containing pancreas tumor from the training dataset of the Medical Segmentation Decathlon challenge [5, 6, 35, 219]. We resample all the CT images to the voxel spacing of $1.3 \times 1.3 \times 2.6$ mm following [106]. We randomly split the dataset into 197 cases for training and 84 cases for testing.

4.5.2 Compared Methods and Processing

4.5.2.1 Context labels based on k-means

We compare CoLab with alternative approaches for context label generation, including a context label generation via clustering. Here, we take pixels inside the body masks or brain masks as samples and employ k-means [4] to construct t clusters. We kept t the same with the class number of human-defined anatomy masks. Specifically, we chose $t = 2$ for liver, kidney, colon and pancreas tumor segmentation and $t = 6$ for brain tumor and brain stroke lesion segmentation.

4.5.2.2 Context labels based on dilated masks

We also compare with a baseline using dilated masks of the ROI as the context labels. This idea is somewhat similar to label smoothing [108, 179] where models are trained with blurred ROI labels.

Specifically, we take the soft dilated masks $\mathcal{M}_i$ defined in Eq. 4.8 and set the context probability as $\boldsymbol{p}_i = [1 - \mathcal{M}_i, \mathcal{M}_i]^\intercal$.

4.5.2.3 Context labels predicted with external datasets

We further evaluate and compare to an approach which leverages prior knowledge from other datasets. Specifically, we trained a segmentation model with 20 CT images using data from [132] which contains labels of 14 abdominal organs including liver and kidney. We resample the 20 CT images to the voxel spacing of 1.6×1.6×3.2 mm. We reduce the dependency of the model on longitudinal axis and trained the segmentation model with a patch size of 128×128×32 as the slice numbers differ across datasets. We then apply this segmentation model to the resampled training split of LiTS as well as KiTS, and extract the liver and kidney masks as the automatically generated contextual anatomy masks.

4.5.2.4 Post-processing

We also compare with a common strategy to suppress FP in segmentation based on component-based post-processing, which is widely adopted in many segmentation pipelines [106]. Specifically, we assume there is always one ROI and remove all but the largest region.

4.5.3 Quantitative Results

We summarize the quantitative results of the models trained with different types of context labels on five binary segmentation datasets in Table 4.1 and one multi-class segmentation dataset in Table 4.3. We separate the results according to whether expert knowledge and manual labeling efforts are needed. Taking the manual segmentation results as the ground truth, we calculate DSC, SEN, PRC and HD. We show some segmentation results with CoLab in Fig. 4.2(c) and also summarize all the corresponding context labels in Fig. 4.9. We provide additional examples of context labels in Fig. 4.10.

As discussed in Section 4.3, the baseline segmentation model seems to underfit the heterogeneous background samples resulting in many FPs and decreased precision. We find most evaluated context labels approaches can improve the models overall segmentation accuracy with increased precision. We observe that adding contextual labels consistently benefits model training and improves performance.

4.5.3.1 Performance of baseline methods

K-means based context labels divide the background samples into different classes based on pixel intensity. Those context labels seem to help the segmentation model separating the abdominal organs and bones from background such as fat and air for the abdominal CT images while in the case of brain magnetic resonance imaging (MRI), it may help segmenting the images into different

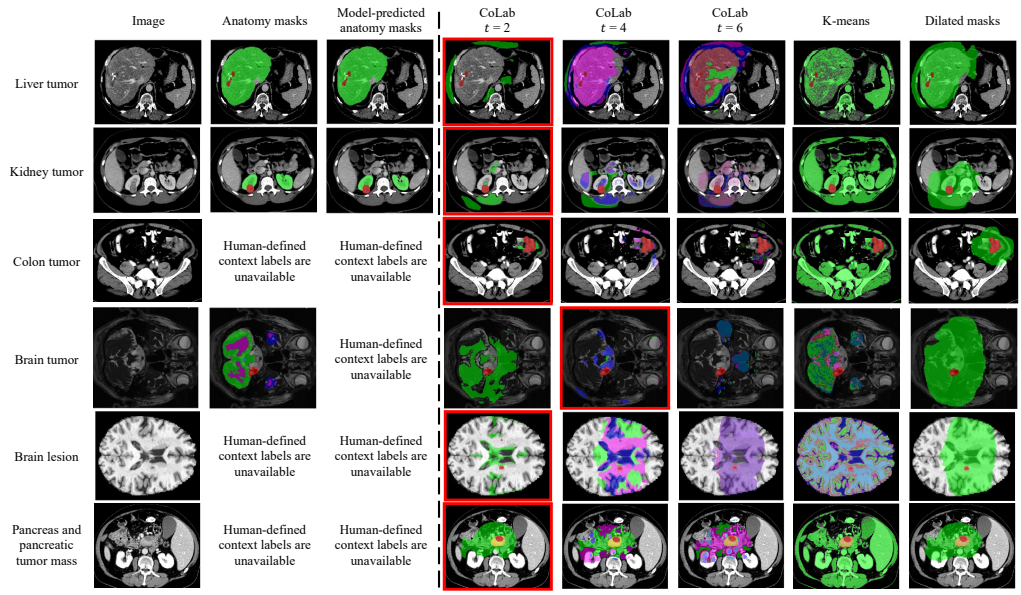


Figure 4.9: Examples of the results $\tilde{f}_\theta(\boldsymbol{x}_i)$ of training data with different types of context labels. The binary ROI masks are shown in red, while multi-class ROI masks are shown in red and orange. The context labels are visualized using other colors The context labels which benefit ROI segmentation most and generated without human input are marked with red boxes. CoLab learns to generate specific structure masks for different tasks. This helps the segmentation network to learn better feaure representations for the background class.

regions such as white matter and ventricles. We find these k-means based context labels are effective for liver and kidney tumor segmentation, while being less effective for colon tumor, brain tumor, brain lesion and pancreas tumor segmentation. This may be due the distinct intensities of large organs and background in CT where the generated labels have meaningful semantic information, benefiting the model training by enhancing the context representations of kidney and liver tumor. Because the appearance of colon tumor is heterogeneous, its segmentation depends more on spatial information and cannot benefit from purely intensity based context labels. As the pancreas label already provides enriched context information for pancreatic tumor segmentation, we find baseline context label techniques such as k-means and dilated masks cannot provide additional benefit. The structure of the brain anatomy in MR is more complex and the classes generated by k-means would be noisy and disconnected. Here, the k-means generated labels are less beneficial for as they provide insufficient context for the ROI.

The context labels based on dilated masks make the segmentation model aware of regions neighbouring the ROIs. We find those context labels are effective for liver tumor, kidney tumor, colon tumor, brain lesion and pancreas segmentation but yield limited improvements for brain tumor segmentation. This may be because liver tumors, kidney tumors, colon tumors and brain lesions appear in varying parts of the primary organ. If trained as binary tasks, the segmentation model

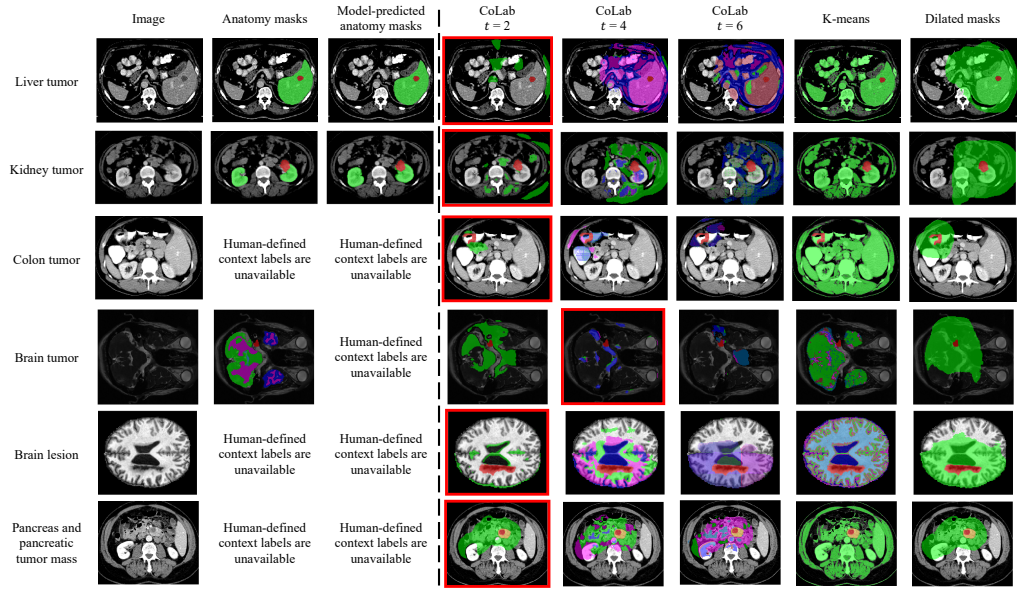


Figure 4.10: Additional examples of the results $\tilde{f}_\theta(\boldsymbol{x}_i)$ of training data with different types of context labels. The binary ROI masks are shown in red, while multi-class ROI masks are shown in red and orange. The context labels are visualized using other colors. The context labels which benefit ROI segmentation most and generated without human input are marked with red boxes.

predicts FP outside the liver, kidney and colon regions or in anatomically unrelated parts of the brain because these samples share similar characteristics with the ROIs, as shown in Fig. 4.9(a). In this case, the dilated masks could enhance the spatial relationship between the ROI and the surrounding context. For example, the dilated masks would connect the liver tumor segmentation with liver context and avoid FPs outside the liver regions. However, the samples of VS dataset only contain one ROI per case and VS always appear around the superior vestibular area. Therefore, the segmentation model does not need more spatial information to locate the ROIs and does not benefit from the dilated masks.

As shown in the third column of Fig. 4.9, we obtain high-quality automatic segmentations of liver and kidney regions as the segmentation of large organs is relatively easy and robust in CT images. The model-predicted masks show similar improvements compared to using the manually labeled anatomy masks. However, this approach is not applicable when we do not have prior knowledge available about the ROIs or when we do not have access to external, manually labelled datasets to train supervised models for context label generation.

4.5.3.2 CoLab

As illustrated in Fig. 4.9(c), CoLab can effectively reduce different types of false positive prediction and improve precision while preserving sensitivity. CoLab can be applied to different tasks

with similar hyper-parameters and show consistent improvements (+2.2 to +8.1 points of DSC). We find CoLab yields better segmentation results than all the other context label approaches that do not use expert knowledge. Moreover, CoLab shows similar improvements compared to using organ masks for liver and kidney tumor segmentation and even better performance for brain tumor segmentation, compared with using tissue masks. It is particularly interesting to find that CoLab can improve the segmentation performance of pancreatic tumor mass, on the top of its strong context label of pancreas. It indicates that there is still room to improve the context representation for segmentation tasks with human defined context labels. CoLab is a flexible and generic method and can be directly applied to boost performance for a variety of segmentation tasks with heterogeneous background classes.

We evaluate CoLab with different number context classes t. We find that when the segmentation model tends to make false positive predictions far from the ROI such as liver tumor, kidney tumor, colon tumor, brain lesion and pancreas tumor, $t = 2$ is the most effective. This may indicate that the segmentation would need the context labels to provide mostly spatial information and $t = 2$ makes CoLab focus on the location of the ROI as the task generator only needs to separate one class of context from the remaining background. This single class would enhance the spatial relationship between the ROI and its surrounding objects. This is consistent with the observations of the results as shown in Fig. 4.9. Specifically, the context labels generated with $t = 2$ highlight the region next to the liver, kidney and colon regions and could implicitly help the segmentation model focus on the vicinity of the ROI boundaries.

When the baseline segmentation model is making false positive predictions around the ROIs such as in the case of VS segmentation, $t = 4$ and $t = 6$ seem more effective. This is probably because the segmentation requires more structured information of surrounding objects and the complicated context of brain anatomy could be better represented when divided into more than 2 classes. As illustrated in Fig. 4.9, context labels generated with $t = 4$ highlight structure information around the ROIs. Specifically, CoLab generates segmentations of the superior vestibular area for VS segmentation where the baseline had predicted FP. Similarly, with the task of brain lesion segmentation, the context labels highlight the regions around ventricles to avoid FP in this area. By doing so, the model learns spatial information by building specific representations for the ventricle regions.

We should note that CoLab may not be very effective with large t, specifically we find CoLab with $t = 6$ would not bring much improvements for kidney tumor, colon tumor and brain lesion segmentation. As shown in Fig. 4.9, we observe that the generated context labels could yield checkerboard patterns. This might be because further decomposing the background class cannot make the task easier to learn. In this case, we find the task generator cannot generate coherent labels and produce similar probabilities for the different classes in those regions, making the context labels trivial and hard to learn.

In general, we find $t = 2$ is most effective for most cases, especially for tumor segmentation in abdominal CT images. Specifically, we suggest the use of $t = 2$ when the baseline model makes false positive predictions distant from the ROIs, and $t = 4$ when over-segmentation is observed closer to the ROIs.

4.5.4 Results with Post-processing

We summarize all the results with component-based post-processing [106] in Table 4.2 and Table 4.4. We find this post-processing would always decrease the segmentation performance for liver tumor, colon tumor and brain lesion as there always exist multiple ROIs per case for those tasks. This indicates that the FPs next to ROIs cannot be eliminated with simple post-processing. In the future, we could consider to utilize Markov random field (MRF) to eliminate FPs for these cases. For kidney tumor, brain tumor and pancreas tumor segmentation, post-processing can improve overall segmentation performance with higher precision for most cases. After post-processing, we find CoLab can help the model achieve the best segmentation performance when compared with other context label generation approaches without human efforts.

4.5.5 Effect on Logit Distributions

We also show the effect of CoLab ($t = 2$ for liver and kidney tumor segmentation and $t = 4$ for brain tumor segmentation) on the logit distributions in Fig. 4.3(c, f, i). We find CoLab has a similar regularization effect than using anatomy masks. By decomposing the background class, the segmentation model can robustly map the background samples to different locations away from the decision boundary, reducing the logit shifts, and thus yielding fewer FPs.

4.5.6 Intensity Histograms of ROIs and Context Labels

We analyze the intensity of context labels of different datasets and summarize the intensity histograms in right part of Fig. 4.11. K-means based context labels are generated based on pixel intensity, therefore the histograms of different classes generally distribute in separate regions. Different classes of dilated masks based context labels always share similar intensity distributions, as dilated masks only takes spatial information into account and does not contain much semantic meanings. CoLab learns to generate context labels within the dilated regions based on meta-learning. Similar to anatomy masks, CoLab would highlight related semantic regions which similar to ROIs but differ from other background samples. The intensity of one of the generated context labels is always similar to that of ROIs. This might indicates that CoLab can generate useful and semantic context labels which can help the segmentation model better distinguish ROIs from similar background samples. We should note that here we only analyze the context labels learned in the final models for CoLab. As the task generator would also be optimized during the training process, it is hard to interpret how CoLab help the segmentation model learn better.

We also look into the intensity histograms of ROIs in left part of Fig. 4.11. We find the intensity histograms of ROIs always largely overlap with that of background samples. The model might require external spatial information to enhance the prior knowledge of ROI location. $t = 2$ could make CoLab highlight the surrounding regions which can help the model reduce false positive predictions on samples with similar intensity but distant from ROIs. In this case, the model cannot benefit from more detailed class representation with $t > 2$ which would further focus on decomposing the samples around ROIs. On the contrary, we notice only the intensity distributions of brain tumor would differentiate from that of background samples. In order to further improve the segmentation accuracy, the model might require more specific subregion information to discriminate between ROIs and the regions with similar intensity. Perhaps for brain tumor segmentation, the model can benefit from knowing the regions which share the most similar intensity distributions with the ROIs, thus requiring more classes of context labels. This can be a potential factor due to which only the optimal context class t of brain tumor segmentation is 4 instead of 2. Practitioners could utilize the intensity histograms of ROI to choose the optimal t. According to these findings, we suggest to choose $t = 2$ for ROIs having similar intensity with the background samples, while choose $t > 2$ for ROIs with distinct intensity distributions from the background samples.

Table 4.1: Evaluation of different binary segmentation tasks with different types of context label generation approaches. We divide the results according to whether expert knowledge and labeling efforts are included. The best and second best results without human efforts are in **bold** with the best also **underlined**.

Task	Method	t	DSC	SEN	PRC	HD
Liver tumor [18]	w/o liver masks	1	54.4	58.8	58.9	111.1
	K-means [4]	2	**61.4**	**61.4**	67.0	71.9
	Dilated masks [179]	2	60.7	59.8	**68.0**	67.6
	CoLab	2	**62.5**	**62.8**	**67.3**	69.4
	CoLab	4	57.3	60.5	62.3	**56.3**
	CoLab	6	59.7	60.3	65.2	**43.6**
	w/ model-predicted liver masks [132]	2	62.4	61.6	70.6	44.1
	w/ liver masks [18]	2	62.8	62.1	69.1	53.5
Kidney tumor [96]	w/o kidney masks	1	74.9	83.2	71.9	120.4
	K-means [4]	2	**76.8**	**83.5**	74.3	87.1
	Dilated masks [179]	2	76.4	**83.9**	73.1	95.3
	CoLab	2	**78.5**	82.2	**77.7**	**75.7**
	CoLab	4	76.4	80.6	**76.5**	**63.7**
	CoLab	6	74.9	81.0	73.3	79.4
	w/ model-predicted kidney masks [132]	2	79.2	81.3	82.7	38.1
	w/ kidney masks [96]	2	79.9	84.1	78.9	54.7
Colon tumor [219]	w/o context labels	1	45.6	**53.9**	45.3	154.9
	K-means [4]	2	44.9	47.2	48.7	111.7
	Dilated masks [179]	2	**47.1**	49.8	**52.8**	**83.3**
	CoLab	2	**48.9**	**53.9**	50.3	86.0
	CoLab	4	45.3	48.1	**51.5**	**81.2**
	CoLab	6	46.1	**50.2**	49.7	88.9
Brain tumor [212]	w/o tissue masks	1	84.3	91.2	80.5	15.2
	K-means [4]	6	85.0	91.3	80.3	9.4
	Dilated masks [179]	2	84.8	91.6	81.1	8.8
	CoLab	2	85.2	90.4	82.5	7.8
	CoLab	4	**89.0**	**91.7**	**86.7**	**1.4**
	CoLab	6	**87.9**	**92.0**	**84.9**	**2.5**
	w/ tissue masks [136, 212]	6	88.2	90.9	86.2	3.1
Brain lesion [152]	w/o context labels	1	58.5	**68.2**	58.6	51.7
	K-means [4]	6	60.0	60.7	**70.5**	**19.5**
	Dilated masks [179]	2	61.9	**69.8**	63.3	35.3
	CoLab	2	**62.8**	**68.2**	**67.4**	28.1
	CoLab	4	**62.6**	68.1	65.4	25.1
	CoLab	6	61.7	66.7	65.7	**24.5**

Table 4.2: Evaluation of different binary segmentation tasks with different types of context labels approaches with post-processing. The results are calculated with component-based post-processing. We divide the results according to whether expert knowledge and labeling efforts are included. The best and second best results without human efforts are in **bold** with the best also **underlined**.

Task	Method	t	DSC	SEN	PRC	HD
Liver tumor [18]	w/o liver masks	1	37.3	29.5	61.6	121.0
	K-means [4]	2	**45.7**	**37.5**	69.9	111.8
	Dilated masks [179]	2	**45.7**	37.3	**71.5**	108.8
	CoLab	2	**48.0**	**40.7**	**70.3**	116.8
	CoLab	4	41.6	35.2	63.7	**101.8**
	CoLab	6	44.0	37.2	66.7	**97.5**
	w/ model-predicted liver masks [132]	2	46.2	37.6	73.4	99.0
	w/ liver masks [18]	2	44.8	36.0	70.2	111.4
Kidney tumor [96]	w/o kidney masks	1	78.2	**79.2**	**79.8**	30.0
	K-means [4]	2	**78.3**	**79.2**	79.3	24.2
	Dilated masks [179]	2	77.5	78.6	77.7	24.2
	CoLab	2	**80.5**	**81.0**	**81.0**	**17.6**
	CoLab	4	76.1	76.5	78.6	**22.7**
	CoLab	6	75.5	76.9	76.2	37.1
	w/ model-predicted kidney masks [132]	2	79.5	78.9	84.7	16.0
	w/ kidney masks [96]	2	80.9	81.5	82.3	17.6
Colon tumor [219]	w/o context labels	1	**48.0**	**48.8**	52.5	**52.6**
	K-means [4]	2	44.3	41.9	50.7	75.2
	Dilated masks [179]	2	46.6	46.4	**54.1**	61.5
	CoLab	2	**47.3**	**46.9**	52.4	53.3
	CoLab	4	44.6	43.5	**54.4**	**51.3**
	CoLab	6	45.1	45.6	49.8	58.0
Brain tumor [212]	w/o tissue masks	1	84.3	87.4	82.0	5.1
	K-means [4]	6	86.9	91.3	83.4	1.6
	Dilated masks [179]	2	83.1	85.7	80.9	4.6
	CoLab	2	84.1	86.5	82.4	4.2
	CoLab	4	**89.0**	**91.7**	**86.7**	**1.4**
	CoLab	6	**88.6**	**92.0**	**86.0**	**1.5**
	w/ tissue masks [136, 212]	6	88.7	90.9	87.0	1.4
Brain lesion [152]	w/o tissue masks	1	57.8	57.1	68.1	25.4
	K-means [4]	6	58.6	56.2	**74.6**	24.6
	Dilated masks [179]	2	**61.8**	**62.2**	67.6	**21.6**
	CoLab	2	61.1	60.7	**72.0**	23.6
	CoLab	4	**61.4**	**61.8**	70.0	**23.0**
	CoLab	6	59.4	58.9	69.4	**21.6**

Table 4.3: Evaluation of pancreas and pancreatic tumor mass segmentation with different types of context label generation approaches. The best and second best results are in **bold** with the best also **underlined**.

Model	t	Pancreas				Tumor mass			
		DSC	SEN	PRC	HD	DSC	SEN	PRC	HD
w/o context labels	1	82.9	82.9	85.1	13.9	48.9	**49.2**	59.3	**52.0**
K-means [4]	2	82.1	82.3	84.6	14.6	44.8	45.4	54.8	64.4
Dilated masks [179]	2	**83.5**	**84.6**	84.4	11.4	48.1	48.2	58.0	58.2
CoLab	2	**83.3**	**83.3**	**85.3**	**7.7**	**51.1**	**50.8**	**63.2**	**44.8**
CoLab	4	82.7	81.7	**86.0**	**7.3**	48.7	45.9	**62.0**	57.3
CoLab	6	82.7	82.9	84.7	8.0	**49.6**	48.4	61.5	52.8

Table 4.4: Evaluation of pancreas and pancreatic tumor mass segmentation with different types of context label generation approaches with post-processing. The results are calculated with component-based post-processing. The best and second best results are in **bold** with the best also **underlined**.

Model	t	Pancreas				Tumor mass			
		DSC	SEN	PRC	HD	DSC	SEN	PRC	HD
w/o context labels	1	82.3	81.9	85.0	9.9	49.0	**48.5**	59.5	**47.6**
K-means [4]	2	83.0	82.3	**86.1**	**7.5**	47.2	45.4	58.3	52.6
Dilated masks [179]	2	**83.7**	**84.1**	85.4	8.1	49.7	48.0	60.1	55.1
CoLab	2	**83.3**	**83.3**	85.5	**7.1**	**51.6**	**50.8**	**63.9**	**41.9**
CoLab	4	82.3	81.3	**85.8**	8.1	47.9	45.2	60.7	61.5
CoLab	6	82.7	82.9	84.8	8.1	**49.8**	48.4	**61.7**	53.7

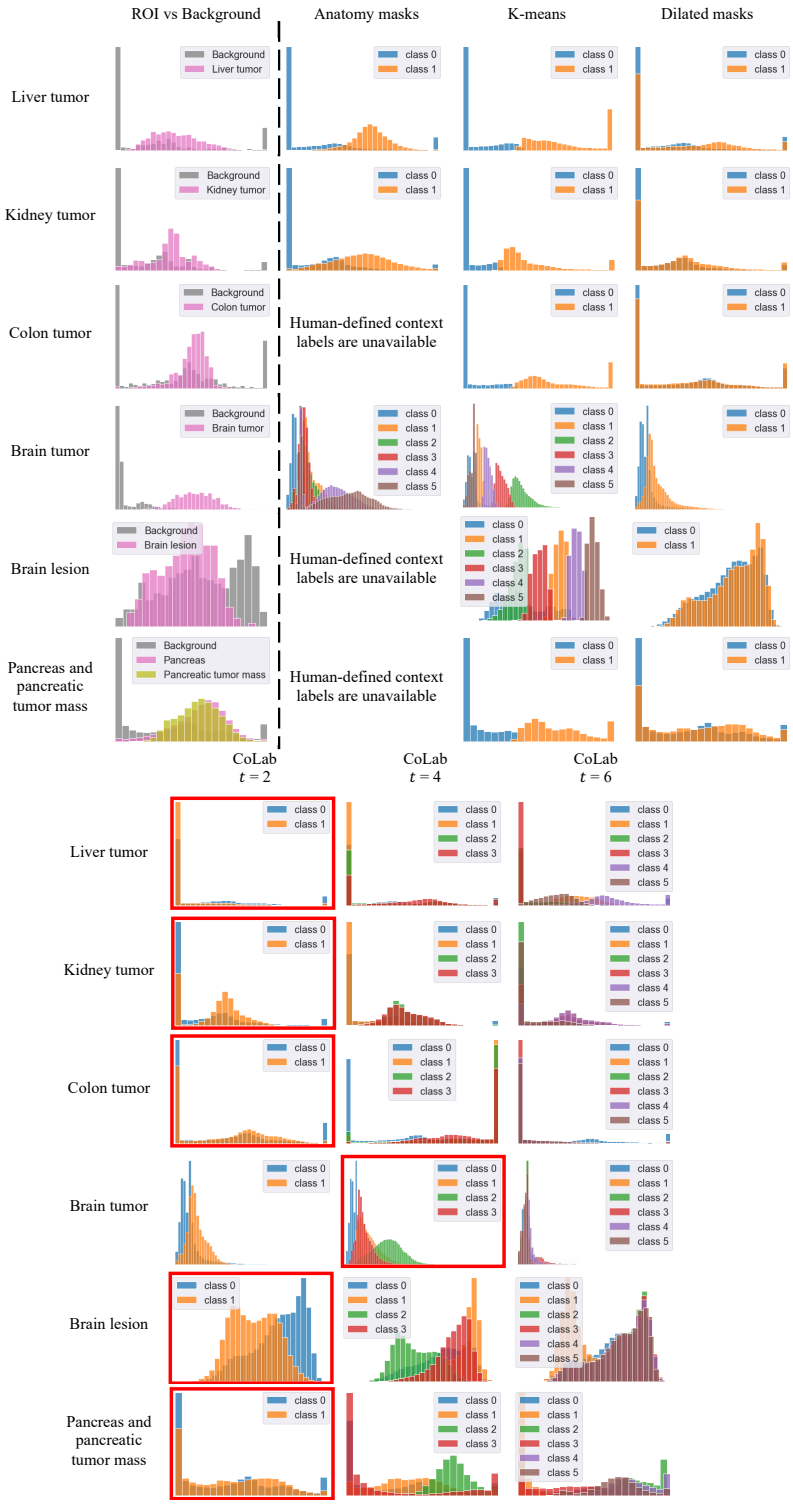


Figure 4.11: The intensity histogram of ROIs calculated based on y_i and the different context labels calculated based on $\tilde{f}_\theta(\boldsymbol{x}_i)$. The histograms of context labels which benefit ROI segmentation most and generated without human input are marked with red boxes.

4.6 Conclusion

In this chapter, we find that segmentation models are prone to over-segment the ROIs in the presence of heterogeneous background classes. With the observation of network behaviour across multiple segmentation tasks, we conclude that this is due to the model being unable to learn discriminative representations from the background samples and the models can be significantly improved by incorporating context labels during training.

We present CoLab, a generic method to automatically generate effective context labels through a meta-learning scheme. We show that CoLab improves overall segmentation performance substantially for several challenging segmentation tasks, without the need of expert knowledge and laborious labeling efforts. Specifically, we find the models trained with CoLab have better segmentation accuracy with higher precision.

As the generated context labels are not always similar to human-defined context labels, we believe the optimal context labels are not unique. In other words, there exist many kinds of context labels that can benefit the learning of heterogeneous background samples. It would be interesting to explore the generalization ability of the generated context labels. Specifically, one can investigate if a kind of generated context labels is beneficial for different tasks.

In future work, we will evaluate CoLab in settings such as domain shifts. It will also be interesting to explore automatic context label generation in multi-task settings where a shared backbone model is trained to automatically identify interesting subclasses across tasks and even datasets.

In the next chapter, we would explore to alleviate the underfitting of background samples from a data distribution perspective. We will focus on improving the heuristic data augmentation to fit the imbalanced dataset better.

Chapter 5

Joint Optimization of Class-Specific Training- and Test-Time Data Augmentation

Data augmentation is crucial for medical imaging segmentation. However, current data augmentation strategies are not always optimal for imbalanced datasets. In Chapter 3 we already shows the effectiveness of adapting different transformations for different categories. The work in this chapter aims to extend that finding and learn the data augmentation policies automatically. The work in this chapter is based on the publications:

1. **Z. Li**, K. Kamnitsas, Q. Dou, C. Qin and B. Glocker. Joint Optimization of Class-Specific Training- and Test-Time Data Augmentation in Segmentation. resubmitted to IEEE Transactions on Medical Imaging.

This work presents an effective and general data augmentation framework for medical image segmentation. We adopt a computationally efficient and data-efficient gradient-based meta-learning scheme to explicitly align the distribution of training and validation data which is used as a proxy for unseen test data. We improve the current data augmentation strategies with two core designs. First, we learn class-specific training-time data augmentation (TRA) effectively increasing the heterogeneity within the training subsets and tackling the class imbalance common in segmentation. Second, we jointly optimize TRA and test-time data augmentation (TEA), which are closely connected as both aim to align the training and test data distribution but were so far considered separately in previous works. We demonstrate the effectiveness of our method on four medical image segmentation tasks across different scenarios with two state-of-the-art (SOTA) segmentation models, DeepMedic and nnU-Net. Extensive experimentation shows that the proposed data augmentation framework can significantly and consistently improve the segmentation performance when compared to existing solutions.

5.1 Introduction

Data augmentation is a de facto standard in neural networks and has shown to improve model generalization [233]. According to the stage when its performed, we can divide data augmentation into TRA and TEA. TRA aims to increase the variation captured by the training dataset by adding perturbed samples with the goal to effectively capture the unseen test data distribution. TEA robustifies the final prediction by averaging predictions of predefined, assumed non-causal variations of test data, to which the model should be robust [236]. An alternative approach for TEA is to modify the test data to achieve higher accuracy with the pretrained model by transforming the test samples to match the distribution of the training data, which is the opposite direction of TRA. Here, we consider to make TRA and TEA complement each other towards the goal of more accurate and robust predictions.

Common data augmentation strategies are usually designed based on heuristics and manually tuned configurations with respect to reducing validation error [106, 117]. However, strategies designed for one task may not be optimal for another task or dataset. Consequently, data augmentation without considering data and task characteristics may not always improve the model performance. In particular, different medical image segmentation tasks may require different data augmentation settings, due to changes in image acquisition protocols, modalities, and anatomical structures of interest [106]. It is tedious to hand-engineer suitable augmentation strategies for each individual task. Therefore, methods have been proposed to automatically learn effective augmentations directly from the available training data [52, 124, 145, 153, 211, 247, 248].

However, we argue that there are two major limitations constraining the performance of current handcrafted and data-driven data augmentation scheme. First, previous studies [52, 124, 153, 211] mostly focus on either TRA or TEA separately, without considering their connections, despite the two being closely linked. This could lead to suboptimal results as the test condition can be adapted through TEA which is not taken into account by TRA when the two are considered in isolation. Second, most TRAs adopt the same transformations for all the samples without considering the different properties existing in different classes. Specifically, the foreground samples in segmentation are more prone to overfitting than background samples and are underrepresented due to class imbalance [147]. Current data augmentation strategies fail to model the heterogeneity of samples from different classes and the resulting model performance may suffer from overfitting under class imbalance.

In this study, we aim to bridge the gap between TRA and TEA by presenting a gradient-based meta-learning framework to automatically discover optimal TRA and TEA strategies, simultaneously. As illustrated in Fig. 5.1(a,b,c,d), data augmentation improves model generalization by making the training and underlying test data distribution overlap more. Specifically, TRA would make the training distribution much wider to better cover the test distribution. Our data augmentation framework

(c.f. Fig. 5.1(e, f)) further takes class properties and test condition into account, fundamentally restructuring the data distributions aiming for an increased overlap.

The contributions of this work can be summarized as follows: 1) We build a bridge between TRA and TEA through joint optimization of data augmentation policies during the training process, which increases the intersection of training and test sample distributions and yields better generalization. 2) We introduce a method that automatically finds different TRA policies for training samples from different classes, implicitly addressing the class imbalance problem. 3) We design a transformation set for TRA with 15 cascaded transformations and 47 operations in total, as well as a transformation set with 83 operations for TEA. These transformation sets cover most transformations in medical image segmentation and can also be easily extended and applied to other applications. 4) Extensive experiments performed on four datasets with two state-of-the-art segmentation models show that our method can consistently improve segmentation performance in various applications and demonstrate the potential to replace the heuristically chosen augmentation policies currently used in most previous works.

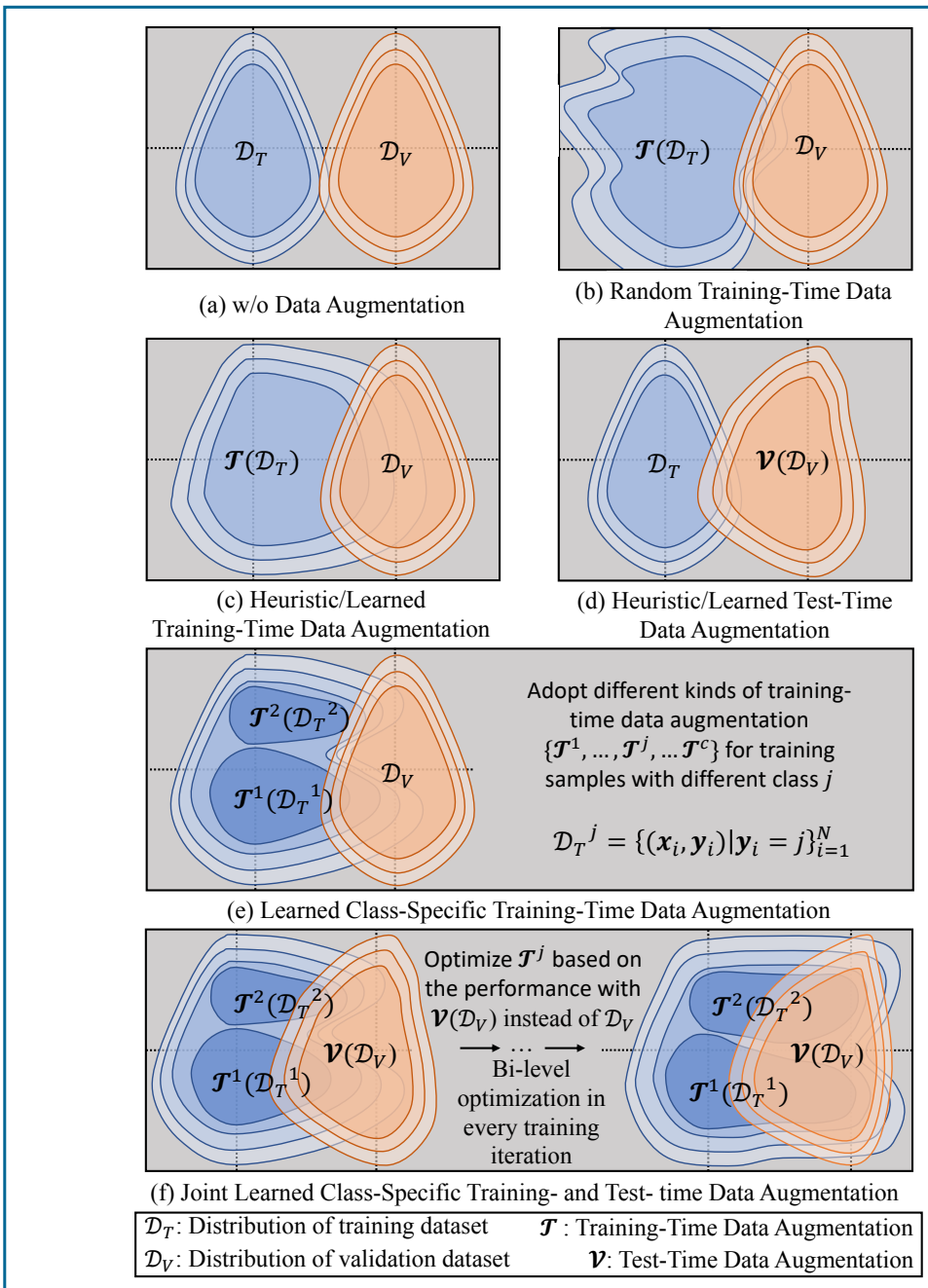


Figure 5.1: Data augmentation improves model performance by aligning the training and validation/test data distribution. As illustrated in (b) [54], (c)[52, 106, 117, 145, 153] and (d) [124, 211], current methods optimize the data distributions by using a validation set as a proxy for unseen test data. Our framework improves the data augmentation by integrating two conceptually simple and intuitive ideas: (e) We adopt different kinds of training-time data augmentation (TRA) for training samples from different classes, effectively extending the training data distribution and alleviating the class imbalance issue. (f) We jointly optimize TRA and test-time data augmentation (TEA) during every training iteration, making the data distributions overlaps more.

5.2 Related Works

5.2.1 Data Augmentation Model

The majority of data augmentation strategies consist of a set of transformations defined based on domain knowledge to represent the heterogeneity of the test data. Examples include rotations, flipping, and intensity shifts [49, 127]. On the other hand, there are also heuristic perturbation techniques such as cutout [60] and mixup [261] that, even though they lead to unrealistic synthetic samples, have been empirically found to improve model generalization. More realistic transformations can be generated based on properties matching [263] or generative adversarial networks [85]. Although these techniques have promising performance, the design of data augmentation is difficult because it requires prior knowledge about the task at hand. Optimal strategies, however, may differ significantly between different tasks, datasets and types of input modalities, and thus are difficult to hand engineer [52, 106]. In this study, we aim to automate the process of designing data augmentation.

Currently most TRA methods adopt the same transformations to all the training samples except [147], which proposed to increase the variance of foreground samples by heuristically reducing the number of transformed samples for the background classes in order to alleviate class imbalance. However, they found different hyper-parameters are optimal for different datasets, and the chosen transformations and hyper-parameters were based on heuristics. In contrast, our method automatically learns different transformations for different classes and discovers the rules from the training data by itself.

5.2.2 Learning Based Training-Time Data Augmentation

There have been many attempts to optimize TRA along with the training process to obtain task-specific TRA policies. Most of the studies are developed based on the idea of adversarial training [79]. The basic adversarial augmentation might not improve generalization on real data as the constructed samples are not realistic. Recent methods attempt to improve real data heterogeneity by adopting an advanced augmentation model [42] or restricting the search space [190], which require strong prior knowledge. Different from these, some methods were proposed to generate artificial samples with task constraints [34], which encourage a generative model to produce additional well-classified images with class properties to enlarge the training data distribution. However, the well-classified samples might not be very useful when the training data is sufficient as they would not make significant changes to the learning of the decision boundary.

Our method is closely related to the line of research which optimizes TRA based on the validation performance such that the learned model can best generalize [52]. Those methods find the sets of augmentation policies that are optimal for a specific training database, out of a pool of possible

transformations, based on reinforcement learning [52, 248], meta-learning [145, 247], or density matching [153]. In our study, we consider to learn the parameters of a probability distribution over TRA with a meta-learning scheme. The meta-learner parameters are optimized with the aim of enabling the task segmentation network to perform better on a validation set. In this way, the meta-learner is explicitly trained to select augmentations that improve generalization. Our method improves existing solutions by the joint optimization of TRA and TEA as well as learning a separate augmentation per class. In addition, the defined transformation pool in our work is more comprehensive than previous studies for medical image segmentation, making it feasible to improve upon current heuristic solutions.

5.2.3 Learning Based Test-Time Data Augmentation

In TEA, class-posterior probabilities from multiple predictions are averaged after applying predefined transformations to the test sample, which is found to be effective to improve accuracy. Recently, some methods were proposed to learn TEA by choosing the transformations obtaining low loss values on the validation set based on a pre-trained model [124, 211]. Test-time adaptation is another kind of learning based TEA where the pre-trained model is adapted to fit a single test sample based on denoising autoencoder [119] or self-supervision [95]. These learning based TEA strategies can be seen as a post-processing to the segmentation and do not contribute to the learning of the model. In contrast, our method proposes to combine the optimization of TRA and TEA during training, which leads to not only learning the optimal TRA and TEA transformations that complement each other, but also learning optimal model parameters given the specific set of data transformations.

5.3 Method

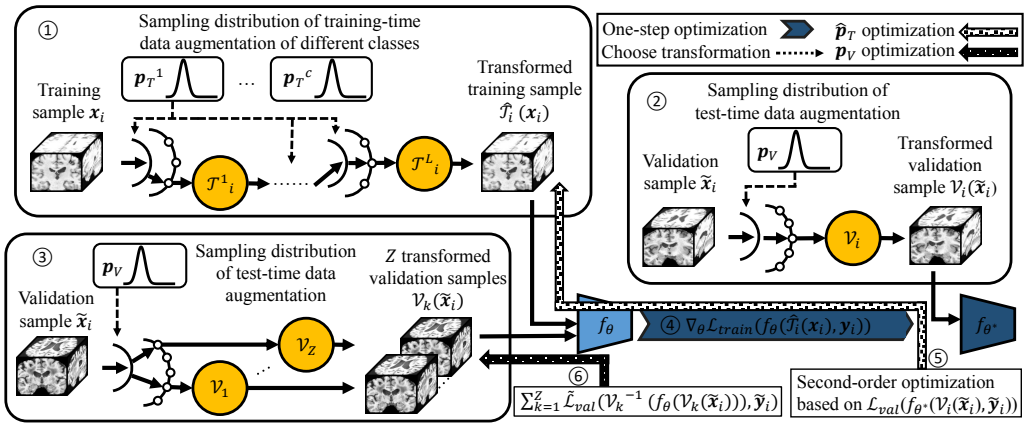


Figure 5.2: The optimization process of the proposed method. In this study, data augmentation is formulated as the probability distribution of multiple predefined transformations, as demonstrated in ①, ② and ③. During the same iteration, class-specific TRA is optimized based on meta-gradients with ⑤ while TEA is optimized based on the validation losses of Z transformed samples with ⑥.

5.3.1 Preliminaries

We consider the image segmentation problem with c total number of classes. A training dataset $\mathcal{D}_T = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$ with N samples is given, where $\boldsymbol{x}_i$ is a training image and $\boldsymbol{y}_i$ corresponds to the segmentation label map with individual labels $y_{ip} \in \{1, ..., c\}$ for each image pixel p. Assuming a segmenter f_θ parameterized by θ, our aim is to learn optimal θ^* parameters, such that $f_{\theta^*}(\cdot)$ minimizes the empirical risk over the training data. For any training loss function $\mathcal{L}_{train}$, the empirical risk of the segmentation model f_θ is defined as $R_{\mathcal{L}_{train}}(f_\theta) = \frac{1}{N}\sum_{i=1}^N \mathcal{L}_{train}(f_\theta(\boldsymbol{x}_i), \boldsymbol{y}_i)$. Apart from $\mathcal{D}_T$, we usually have a validation dataset $\mathcal{D}_V = \{(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{y}}_i)\}_{i=1}^M$ with M samples along with a validation loss $\mathcal{L}_{val}$, which is taken as a proxy for unseen test data and used to tune the hyper-parameters including learning rates [140], network architecture [272] and data augmentation policies [52]. Note that $\mathcal{D}_V$ could come from a different distribution from $\mathcal{D}_T$, based on different assumptions of unseen test data.

5.3.2 Sampling Transformations

For a sample $\boldsymbol{x}_i$ (or $\tilde{\boldsymbol{x}}_i$), we will apply transformation $\mathcal{T}_i(\cdot)$ which is specific to the i-th sample. $\mathcal{T}_i$ is obtained by sampling from a set of K operations $\{\mathcal{O}^1, ..., \mathcal{O}^K\}$ based on the corresponding probability distribution $\boldsymbol{p} = [p_1, ..., p_K]^\intercal$. In this study we represent with a different $\mathcal{O}^j$ operation not only transformations of different type (for example rotations, contrast enhancement, etc.) but also transformations of the same type but different magnitudes (for example rotations of different

degree). We do not further optimize the predefined magnitudes of transformations during training. Our method will optimize during training the sampling distribution $\boldsymbol{p}$ of different transformations for data augmentation, so that we learn which transformations are most appropriate for the given dataset and task.

In order to include the distribution into the gradient based optimization through the non-differentiable sampling process, we reparameterize the categorical distribution using the Gumbel-Softmax trick [110]. We calculate the probability of assigning sample $\boldsymbol{x}_i$ (or $\tilde{\boldsymbol{x}}_i$) with operation $\mathcal{O}^j$ as:

$$s_{ij} = \frac{\mathrm{e}^{p_j + g_{ij}}}{\sum_{v=1}^{K} \mathrm{e}^{p_v + g_{iv}}}, \qquad \text{for } j = 1, ..., K, \tag{5.1}$$

where g_{ij} is a sample drawn from the Gumbel distribution, i.e., $g_{ij} = -\log(-\log(\varepsilon))$, in which ε is a random number by drawing $\varepsilon \sim \text{Uniform}(0, 1)$. It then holds that $\sum_{j=1}^{K} s_{ij} = 1$ and $0 \leq s_{ij} \leq 1, \forall j$. In this way, the stochasticity involved in the sampling process is removed from the computational graph of network's training and the process of choosing augmentation $\mathcal{T}_i$ based on probability distribution $\boldsymbol{p}$ now becomes differentiable. Specifically, $\mathcal{T}_i$ is chosen as $\mathcal{O}^{j^*}$ where $j^* = \mathrm{argmax}_j(s_{ij})$. The sampling probability $\boldsymbol{p}$, which we would like to optimize, can still not be updated via backpropagation, both due to the non differentiable argmax and because the transformations are non-differentiable in the general case. To work around this, we also calculate a weight w_i that corresponds to the sample $\boldsymbol{x}_i$ (or $\tilde{\boldsymbol{x}}_i$) with:

$$w_i = \max_j(s_{ij}) + \underbrace{(1 - \max_j(s_{ij}))}_{\text{does not require gradient}}, \tag{5.2}$$

which is a function of the sampling probability s_{ij} and is always equal to 1. In this way, we can use w_i as a means to optimize $\boldsymbol{p}$ with gradient descend. With this notion, we add the weight into the empirical risk as $R'_{\mathcal{L}_{train}}(f_\theta) = \frac{1}{N} \sum_{i=1}^{N} w_i \mathcal{L}_{train}(f_\theta(\boldsymbol{x}_i), \boldsymbol{y}_i)$. In this manner, w_i and therefore s_{ij} are part of the total loss and hence can be straightforwardly optimized.

For distinction between TRA and TEA, in the following paragraphs we define the probability distribution and transformation of TRA as $\boldsymbol{p}_T$ and $\hat{\mathcal{T}}_i$ while denoting the ones of TEA as $\boldsymbol{p}_V$ and $\mathcal{V}_i$ unless otherwise noted. Note that we are considering the problem of image segmentation, therefore the spatial transformations are always applied to $\boldsymbol{y}_i$ simultaneously but we omit this for simplicity.

5.3.3 Learning Scheme

We describe the high-level learning scheme of the proposed method in Fig. 5.3. During an optimization process (commonly stochastic gradient descent for neural networks), we minimize the

$\mathcal{L}_{train}$ over all training data and yield the learned parameters θ^{T*}. In practice, the learned θ^{T*} is sub-optimal for validation data because the training data cannot cover all the underlying data properties. Data augmentation is widely utilized to implicitly reduce the generalization gap $\theta^{T*} \rightarrow \theta^{V*}$ by using additional artificial training samples $\boldsymbol{\mathcal{T}}(\mathcal{D}_T)$. However, it is not guaranteed that $\boldsymbol{\mathcal{T}}(\cdot)$ is always effective during this process.

We propose to explicitly close the generalization gap by aligning the training and test data distribution. We optimize the model parameter from θ^{T*} to $\theta^{\tilde{T}*}$ by learning a TRA model with a meta-learning scheme. Meanwhile, we transform the validation data in the way it can be easier to be recognized and change the target optimal parameters of validation from θ^{V*} to $\theta^{\tilde{V}*}$. By the joint optimization of TRA and TEA, we are able to close the generalization gap from $\theta^{T*} \rightarrow \theta^{V*}$ to $\theta^{\tilde{T}*} \rightarrow \theta^{\tilde{V}*}$.

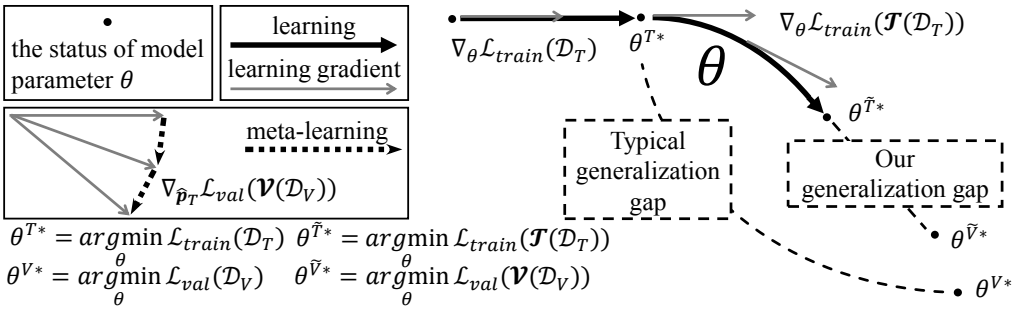


Figure 5.3: The learning scheme of the proposed method. When we train the model with the training data $\mathcal{D}_T$ and the training criterion $\mathcal{L}_{train}$, the model does not always generalize well on the validation data $\mathcal{D}_V$. We aim to close this generalization gap explicitly by the joint optimization of TRA $\boldsymbol{\mathcal{T}}(\cdot)$ and TEA $\boldsymbol{\mathcal{V}}(\cdot)$.

5.3.4 Overview of the Training Process

We aim to reduce the generalization gap explicitly by optimizing a probability distribution over data augmentations $\boldsymbol{p}_T$ (or $\boldsymbol{p}_V$) based on the gradient from the validation data $\nabla_{\boldsymbol{p}_T}\mathcal{L}_{val}$ (or $\nabla_{\boldsymbol{p}_V}\mathcal{L}_{val}$). Thus, $\boldsymbol{p}_T$ (or $\boldsymbol{p}_V$) is automatically adapted to the underlying task-specific characteristics.

We formulate the TRA policy as the meta-learning parameters, and similar to our solutions for context label learning (CoLab) as described in Section 4.4. We develop a training framework based on meta-learning via second-order optimization to accomplish this. The optimization process of the proposed method is illustrated in Fig. 5.2. During the sampling process, we first obtain the transformed training data with ① and optimize the model to f_{θ^*} with ④ based on a single optimization step; then we pass the transformed validation data with ② through f_{θ^*} to compute the second-order gradients ⑤ and backprop to ①, to learn TRA that leads to learning θ^* that best generalizes on validation data transformed with TEA; meanwhile, we also apply varied transformations of TEA to a single validation sample with ③ and we try to learn TEA $\boldsymbol{p}_V$ which can transform a single

validation sample $\tilde{\boldsymbol{x}}_i$ to have the lowest validation error with ⑥.

Algorithm 2 Joint Optimization of Class-Specific Training- and Test-Time Data Augmentation in Segmentation

Require:
$\mathcal{D}_T = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$: training data, $\mathcal{D}_V = \{(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{y}}_i)\}_{i=1}^M$: validation data; $f_\theta(\cdot)$: the segmentation model, $\hat{\mathcal{T}}_i(\cdot)$: TRA which is determined by drawing from class-specific probability $\hat{\boldsymbol{p}}_T$, $\mathcal{V}_i(\cdot)$: TEA which is determined by drawing from probability $\boldsymbol{p}_V$.
α, β, γ: learning rate to update θ, $\hat{\boldsymbol{p}}_T$ and $\boldsymbol{p}_V$.

Initialize $\hat{\boldsymbol{p}}_T, \boldsymbol{p}_V$ with heuristic policies referring to the ones in DeepMedic [117] or nnU-Net [106].
for each iteration **do**
  Sample a batch of training data $\mathcal{B}_T = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$ from $\mathcal{D}_T$ and a batch of validation data $\mathcal{B}_V = \{(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{y}}_i)\}_{i=1}^m$ from $\mathcal{D}_V$.
  for a number of steps **do** ▷ *Note: One step is sufficient in our experiments.*
    Sample a set of $\{\hat{\mathcal{T}}_i(\cdot)\}_{i=1}^n$ with Gumbel-Softmax distribution parameterized by ${\boldsymbol{p}_T}^j$ based on sample class.
    Sample $\{\mathcal{V}_i(\cdot)\}_{i=1}^m$ and $\{\mathcal{V}_k(\cdot)\}_{k=1}^Z$ with Gumbel-Softmax distribution parameterized by $\boldsymbol{p}_V$.
    Calculate θ^* with an optimization step via Eq. 5.6.
    Optimize $\hat{\boldsymbol{p}}_T$ based on normalized meta-gradients:
    $\hat{\boldsymbol{p}}_T^{t+1} = \hat{\boldsymbol{p}}_T^t - \beta \nabla_{\hat{\boldsymbol{p}}_T^t} \frac{1}{m} \sum_{i=1}^m \mathcal{L}_{val}(f_{\theta^*}(\mathcal{V}_i(\tilde{\boldsymbol{x}}_i)), \tilde{\boldsymbol{y}}_i)$. ▷ *Learning of TRA.*
    Optimize $\boldsymbol{p}_V$ based on normalized gradient via Eq. 5.12. ▷ *Learning of TEA.*
  end for
  Update θ to θ^*. ▷ *Training the segmentation model.*
end for

5.3.5 Learning of Class-Specific Training-Time Data Augmentation

5.3.5.1 The design of predefined transformations

Following the design of data augmentation in many medical image segmentation frameworks [106, 117], we design the transformation set with L=15 cascaded operations including rotation, mirroring, gamma correction, histogram transformations, blurring, sharpening, adding noise, and simulating low resolution. The operation magnitudes are decided by uniformly sampling from predefined ranges. We summarize the detailed information about the operations in Table 5.1. Specifically, the probability distribution and transformations of TRA is extended as $\boldsymbol{p}_T = (\boldsymbol{p}^1, ..., \boldsymbol{p}^L)$ and $\hat{\mathcal{T}}_i = \{{\mathcal{T}^1}_i, ..., {\mathcal{T}^L}_i\}$.

We ensure that our design of TRA is able to accomplish the same functionality with the built-in data augmentation in prevailing frameworks such as DeepMedic [117] and nnU-Net [106], therefore our method can act as a replacement for heuristic TRA. We initialize $\boldsymbol{p}_T$ with heuristic policies provided by these frameworks, as shown in Fig. 5.7.

5.3.5.2 Class-specific data augmentation

We adopt different TRAs for training samples from different classes. Specifically, we extend the probability distribution to $\hat{\boldsymbol{p}}_T = ({\boldsymbol{p}_T}^1, ..., {\boldsymbol{p}_T}^c)$ which contains different probability distributions

for c classes. In this way, TRA becomes more flexible and powerful as it gains the ability to draw $\hat{\mathcal{T}}_i$ from different distributions for different classes.

In practice, we determine the class of a training patch with the central pixel of the patch. Note that in this study we only regard the training samples to come from 2 classes consisting of foreground (tumor, lesion, and organs) and background.

5.3.5.3 Policy optimization with meta-gradients

Similar to previous works on learning TRA [52, 145], we aim to learn TRA based on the performance of validation data and formulate the optimization of TRA as a bi-level optimization problem:

$$\min_{\hat{\boldsymbol{p}}_T} \frac{1}{M} \sum_{i=1}^{M} \mathcal{L}_{val}(f_{\theta^*}(\tilde{\boldsymbol{x}}_i), \tilde{\boldsymbol{y}}_i) \tag{5.3}$$

$$s.t. \quad \theta^* = \underset{\theta}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{train}(f_{\theta}(\hat{\mathcal{T}}_i(\boldsymbol{x}_i)), \boldsymbol{y}_i). \tag{5.4}$$

We propose to solve this based on gradient descent following [74, 192]. We train the model with a training batch containing n samples and a validation batch consisting of m samples. For simplicity, we shorten $\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_{train}(f_{\theta}(\hat{\mathcal{T}}_i(\boldsymbol{x}_i), \boldsymbol{y}_i))$ as $\mathcal{L}_{train}(\theta, \hat{\boldsymbol{p}}_T)$ and $\frac{1}{m}\sum_{i=1}^{m} \mathcal{L}_{val}(f_{\theta^*}(\tilde{\boldsymbol{x}}_i), \tilde{\boldsymbol{y}}_i))$ as $\mathcal{L}_{val}(\theta^*)$ in the following paragraphs. Based on the chain rule, the gradient of validation loss w.r.t. $\hat{\boldsymbol{p}}_T$ is derived as:

$$\nabla_{\hat{\boldsymbol{p}}_T} \mathcal{L}_{val}(\theta^*) = (\frac{\partial \theta^*}{\partial \hat{\boldsymbol{p}}_T})^{\intercal} \nabla_{\theta} \mathcal{L}_{val}(\theta^*), \tag{5.5}$$

where $\nabla_{\hat{\boldsymbol{p}}_T} = (\frac{\partial}{\partial \hat{\boldsymbol{p}}_T})^{\intercal}$ and $\nabla_{\theta} = (\frac{\partial}{\partial \theta})^{\intercal}$. The calculation of $\frac{\partial \theta^*}{\partial \hat{\boldsymbol{p}}_T}$ can be derived based on implicit function theorem [15]. However, the calculation would introduce a Hessian which is not practical to calculate with the parameters of deep nerual network as the number of parameters is too large. There are many methods to approximate the gradient without Hessian calculation [74, 157, 192], in this study we choose to approximate θ^* by using a single training step [73, 157]. Specifically, we approximate the optimal θ^* via a standard training step with:

$$\theta^* \approx \theta - \alpha \nabla_{\theta} \mathcal{L}_{train}(\theta, \hat{\boldsymbol{p}}_T). \tag{5.6}$$

Here, α is the step length which we set equal to the learning rate of the task model. Eq. 5.6 evaluates the effectiveness of the sampled data augmentation. We differentiate this equation w.r.t. $\hat{\boldsymbol{p}}_T$ from both sides and yield:

$$\frac{\partial \theta^*}{\partial \hat{\boldsymbol{p}}_T} = -\alpha \nabla^2_{\theta, \hat{\boldsymbol{p}}_T} \mathcal{L}_{train}(\theta^*, \hat{\boldsymbol{p}}_T), \tag{5.7}$$

where $\nabla^2_{\theta,\hat{\boldsymbol{p}}_T} = \frac{\partial \nabla_\theta}{\partial \hat{\boldsymbol{p}}_T}$. By substituting Eq. 5.7 into Eq. 5.5, now we can update $\hat{\boldsymbol{p}}_T$ with:

$$\begin{aligned} \hat{\boldsymbol{p}}_T^{t+1} &= \hat{\boldsymbol{p}}_T^t - \beta \nabla_{\hat{\boldsymbol{p}}_T^t} \mathcal{L}_{val}(\theta^*) \\ &= \hat{\boldsymbol{p}}_T^t + \alpha\beta \nabla^2_{\hat{\boldsymbol{p}}_T^t, \theta} \mathcal{L}_{train}(\theta^*, \hat{\boldsymbol{p}}_T^t) \nabla_\theta \mathcal{L}_{val}(\theta^*), \end{aligned} \tag{5.8}$$

which can be interpreted as a gradient of the gradient from the task-driven training. In the above, β is the learning rate for determining the probability distribution. In this way, we can optimize the distribution $\hat{\boldsymbol{p}}_T$ explicitly with the aim to improve generalization of the segmentation model using the validation data. Eq. 5.8 includes a second-order gradient. As the distribution $\hat{\boldsymbol{p}}_T$ is represented with only a few parameters (K, which is in the order of 10-100), we find the complexity of the gradient computation to be $O(|\hat{\boldsymbol{p}}_T||\theta|)$ which is feasible and can be handled by prevailing toolboxes such as PyTorch and Tensorflow.

After updating $\hat{\boldsymbol{p}}_T$, we update f_θ to f_{θ^*} to fit the updated TRA policy for the next iteration. We optimize $\hat{\boldsymbol{p}}_T$ along with the training of the task model, and at the end of the training we may have higher performance than any model learned with a random or manually configured augmentation policy.

5.3.5.4 Gradient normalization

We normalize the gradient from different classes of training samples, as we notice that the contributions of training samples from different classes to the reduction of the validation loss varied a lot. For example, the foreground samples are more effective for reducing the validation loss, resulting in increased probabilities of the policies associated with the foreground samples. Specifically, if we rewrite the gradient in Eq. 5.5 by the chain rule as:

$$\nabla_{\hat{\boldsymbol{p}}_T} \mathcal{L}_{val}(\theta^*) = \sum_{i=1}^{n} \nabla_{w_i} \mathcal{L}_{val}(\theta^*) (\frac{\partial w_i}{\partial \hat{\boldsymbol{p}}_T})^\intercal, \tag{5.9}$$

we would find the magnitude of $\nabla_{w_i} \mathcal{L}_{val}(\theta^*)$ is significantly larger for the foreground samples than the background samples. To resolve this, we rewrite Eq. 5.9 as $\nabla_{\hat{\boldsymbol{p}}_T} \mathcal{L}_{val}(\theta^*) = \sum_{i=1}^{n} h_i (\frac{\partial w_i}{\partial \hat{\boldsymbol{p}}_T})^\intercal$ with the normalized gradient h_i:

$$h_i = \nabla_{w_i} \mathcal{L}_{val}(\theta^*) - \frac{\sum_{j=1}^{n} \mathbb{1}_{[y_{jc}=y_{ic}]} \nabla_{w_j} \mathcal{L}_{val}(\theta^*)}{\sum_{j=1}^{n} \mathbb{1}_{[y_{jc}=y_{ic}]}} \tag{5.10}$$

where y_{ic} is the central pixel label of the segmentation label map $\boldsymbol{y}_i$ and $\mathbb{1}_{y_{jc}=y_{ic}} \in \{0, 1\}$ is an indicator function which is equal to 1 if and only if $y_{jc} = y_{ic}$. Thus, the gradients are normalized for different classes. Another benefit of gradient normalization is that we can guarantee that the probability of transformation which is not sampled in one iteration would remain unchanged.

5.3.5.5 Sampling normalization

We notice that the optimization process could also be biased towards transformations with high probability. Because the more frequently one transformation is sampled, its probability would be increased more as long as it is more effective than the majority of the transformations in the same batch. As a consequence, it would be likely to be trapped in a local minimum and the probability of any preferable transformation which is frequently sampled by chance could be increased a lot. Therefore, we also normalize h_i with the sampling frequency and obtain $\hat{h}_i$ as:

$$\hat{h}_i = \frac{h_i}{\sum_{v=1}^{n} \mathbb{1}_{[\text{argmax}_j(s_{vj})=\text{argmax}_j(s_{ij})]}}. \tag{5.11}$$

5.3.6 Learning of Test-Time Data Augmentation

5.3.6.1 The design of predefined transformations

We design the transformation set for TEA with K=84 kinds of deterministic operations including identity along with 41 spatial transformations, 30 intensity transformations, and 12 noise transformations. For TEA, we use a single transformation for a test data, which is a common practice for current solutions. Concatenated operations are rarely utilized and might not be necessary for this setting. This might be because we cannot adopt too many TEA transformations, which will largely increase the inference time.

We summarize the detailed information about those transformations in Table 5.1. We make sure the test-time data augmentation is realistic by restricting the predefined parameters of the operations from which the algorithms can choose. We initialize $\boldsymbol{p}_V$ referring to the heuristic policies used in nnU-Net [106] which comprises mirroring and $180°$ rotation in three directions, as shown in Fig. 5.7.

5.3.6.2 Policy optimization based on reverted predictions

The optimization of TEA is straightforward. We update $\boldsymbol{p}_V$ by choosing the transformations which have the lowest validation loss with the same validation sample:

$$\boldsymbol{p}_V^{t+1} = \boldsymbol{p}_V^t - \gamma \nabla_{\boldsymbol{p}_V} \frac{1}{Z} \sum_{k=1}^{Z} \tilde{\mathcal{L}}_{val}(\mathcal{V}_k{}^{-1}(f_\theta(\mathcal{V}_k(\tilde{\boldsymbol{x}}_i))), \tilde{\boldsymbol{y}}_i), \tag{5.12}$$

where γ is the learning rate to update the probability and $\tilde{\mathcal{L}}_{val}$ is the validation loss function for TEA optimization, which can differ from $\mathcal{L}_{val}$.

5.3.6.3 Sampling normalization

Similar to TRA, we notice that the optimization of $\boldsymbol{p}_V$ would be biased due to the sampling results. We simplify the second term on the right of Eq. 5.12 to $\nabla_{\boldsymbol{p}_V}\tilde{\mathcal{L}}_{val}(\theta)$, and derive the gradient based on the chain rule:

$$\nabla_{\boldsymbol{p}_V}\tilde{\mathcal{L}}_{val}(\theta) = \sum_{k=1}^{Z}\tilde{h}_k(\frac{\partial w_k}{\partial \boldsymbol{p}_V})^\intercal. \tag{5.13}$$

Similarly, we normalize the gradients based on sampling frequency and calculate $\tilde{h}_k$ as:

$$\tilde{h}_k = \frac{\nabla_{w_k}\tilde{\mathcal{L}}_{val}(\theta)}{\sum_{v=1}^{Z}\mathbb{1}_{[\text{argmax}_j(s_{vj})=\text{argmax}_j(s_{kj})]}}. \tag{5.14}$$

5.3.7 Inference of Test-Time Data Augmentation

Given unseen test data, we adopt the learned TEA policy to transform the image. Specifically, in order to simplify the inference process, we do TEA at test time with the weighted sum of operations that have the highest z probability (z=8 for three-dimensional (3D) U-Net and z=4 for DeepMedic). The weight of operation $\mathcal{O}^j$ is set as the corresponding sampling probability which is calculated as $\mathrm{e}^{p_j}/\sum_{v=1}^{K}\mathrm{e}^{p_v}$.

5.3.8 Joint Learning of Training- and Test-Time Data Augmentation

We propose to jointly optimize TRA and TEA, and specifically we optimize $\hat{\boldsymbol{p}}_T$ with the transformed validation data based on $\boldsymbol{p}_V^t$ and rewrite Eq. 5.3 and 5.4 as:

$$\min_{\hat{\boldsymbol{p}}_T}\frac{1}{M}\sum_{i=1}^{M}\mathcal{L}_{val}(f_{\theta^*}(\mathcal{V}_i(\tilde{\boldsymbol{x}}_i)),\tilde{\boldsymbol{y}}_i) \tag{5.15}$$

$$s.t.\quad \theta^* = \underset{\theta}{\text{argmin}}\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_{train}(f_\theta(\hat{\mathcal{T}}_i(\boldsymbol{x}_i)),\boldsymbol{y}_i). \tag{5.16}$$

Note that we optimize both $\hat{\boldsymbol{p}}_T$ and $\boldsymbol{p}_V$ in one training iteration. Bridging the optimization process of $\hat{\boldsymbol{p}}_T$ and $\boldsymbol{p}_V$ has two advantages: First, we can reduce the risk of overfitting on the validation data as it is extended with augmented samples. Second, the model can generalize well to the transformations we adopt at test time. In this way, TRA and TEA can inform each other during the

learning process. The full procedure is summarized in Algorithm 2. The training process of one iteration is illustrated in detail in Fig. 5.4.

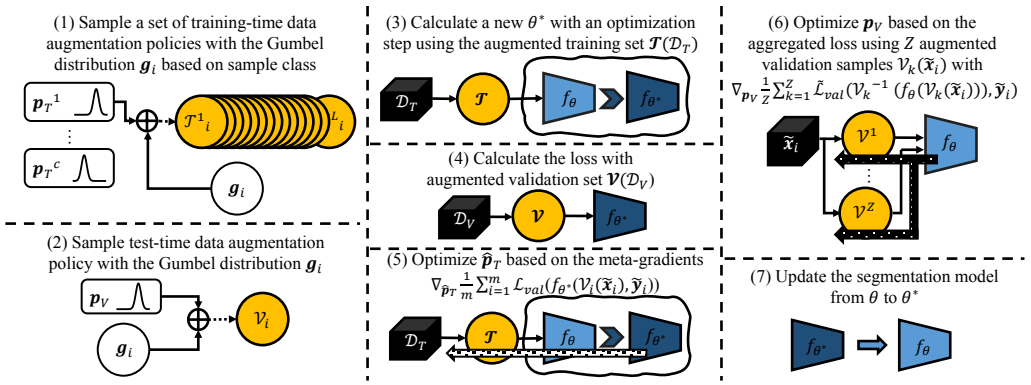


Figure 5.4: The illustration of one iteration during the training process. It is consistent with Algorithm 2.

5.3.9 List of Operations for Training-Time Data Augmentation

We list all the operations $\mathcal{O}$ we use for TRA in Table 5.1. Note that we design most operations with stochastic magnitudes in a symmetric way. That is to say, the transformed image could be transformed back to the original image using the same operation, with the probability of 50%. In this way, we can increase the variance of training dataset with more realistic and potentially useful samples.

Different from the operation set in AutoAugment which includes operations with deterministic magnitudes [52], we design each operation with stochastic magnitudes which are sampled from a uniform sampling probability. In this way, we can cover transformations with larger variance and realize similar functionalities with TRA used in prevailing segmentation methods.

A straightforward question is whether we can further improve TRA by 1) adding the number of predefined transformations, 2) optimizing the magnitude of transformations as well or 3) including more complicated transformations such as generative models (i.e. utilizing a network to generate the augmented samples). In fact, we find that if we extend the transformation set with more choices (i.e. larger K), the performance would not be better. It is because enlarging K would make transformations being sampled less and make TRA harder to optimize. We find 2-10 is a reasonable range for K. It would be feasible to additionally optimize the magnitude of the operations. However, we find that it would make the optimization process unstable when we also optimize the operation magnitudes. This is might because controlling the operation magnitude is hard and the transformed images could become unrealistic. In addition, we are not sure if the function of changing operation magnitude is continuous in reducing the validation loss.

We also try to utilize a neural network to transform the training samples. Nevertheless, our initial experimental results show that the model would easily overfit the validation data but cannot perform well with unseen test data. The transformation model can easily align the training and validation data very well as the transformation model has a large mount of parameters. However, the relationship is too specific and does not generalize well on underlying test data. We observe that reducing model parameters of the transformation models or adding model regularization can alleviate the overfitting issue but it is still hard to achieve similar results with heuristic TRA strategies.

Table 5.1: List of all operations that our method can choose for TRA. We formulate TRA as a composition of L=15 types of operations, with each operation has K choices of magnitudes. K varies from 2 to 7.

Category	Series ID	Operation Name	Description	Range of magnitudes
Spatial transformations	0	Scaling	Scale up/ down the image with factor 1+M	[[0, 0.1), [0.1, 0.2), [0.2, 0.3), [0.3, 0.4), [0.4, 0.5)]
	1	RotateFrontal	Rotate the image along frontal axis anticlockwise/ clockwise	[[0, 10°), [10°, 20°), [20°, 30°), 90°]
	2	RotateSagittal	Rotate the image along sagittal axis anticlockwise/ clockwise	[[0, 10°), [10°, 20°), [20°, 30°), 90°]
	3	RotateLongitudinal	Rotate the image along longitudinal axis anticlockwise/ clockwise	[[0, 10°), [10°, 20°), [20°, 30°), 90°]
	4	MirroringSagittal	Flip the sample in sagittal planes	None
	5	MirroringFrontal	Flip the sample in frontal planes	None
	6	MirroringAxial	Flip the sample in axial planes	None
Intensity transformations	7	Gamma correction	Scale $\boldsymbol{x}$ to [0,1], then $\mathcal{O}(\boldsymbol{x}) = \boldsymbol{x}^{(1+\gamma)^{\pm 1}}$, and scale it back	[[0, 0.2), [0.2, 0.4), [0.4, 0.6)]
	8	Inverted gamma correction	Do the gamma correction with the inverted image	[[0, 0.2), [0.2, 0.4), [0.4, 0.6)]
	9	Shifting intensity histogram	$\mathcal{O}(\boldsymbol{x}) = \boldsymbol{x} \pm$ shift	[[0, 0.1), [0.1, 0.2), [0.2, 0.3)]
	10	Scaling intensity histogram	$\mathcal{O}(\boldsymbol{x}) = \boldsymbol{x}$ * (1 + scale)$^{\pm 1}$	[[0, 0.1), [0.1, 0.2), [0.2, 0.3)]
	11	Contrast	Reduce the image mean, then $\mathcal{O}(\boldsymbol{x}) = \boldsymbol{x}$ * (1 + scale)$^{\pm 1}$, and add the mean value back	[[0, 0.1), [0.1, 0.2), [0.2, 0.3)]
Noise transformations	12	Blurring	Blur the image using Gaussian filter with different standard deviations	[[0.4, 0.6), [0.6, 0.8), [0.8, 1)]
		Sharpening	Sharpen the image using Laplacian of Gaussian filter with different standard deviations	[[0.8, 1), [0.6, 0.8), [0.4, 0.6)]
	13	Adding Gaussian noise	Add Gaussian noise with different standard deviations	[[0, 0.05), [0.05, 0.10), [0.10, 0.15)]
	14	Simulating low resolution	Scale down the image, and scale it back	[[0.8, 1), [0.6, 0.8), [0.4, 0.6)]

5.3.10 List of Transformation for Test-Time Data Augmentation

We list all the operations $\mathcal{O}$ we use for TEA in Table 5.2. Note that the operations always have deterministic magnitudes. In this way, we can make sure we would apply the same set of transformation to a test sample. We show the example the transformations in TEA in Fig. 5.5.

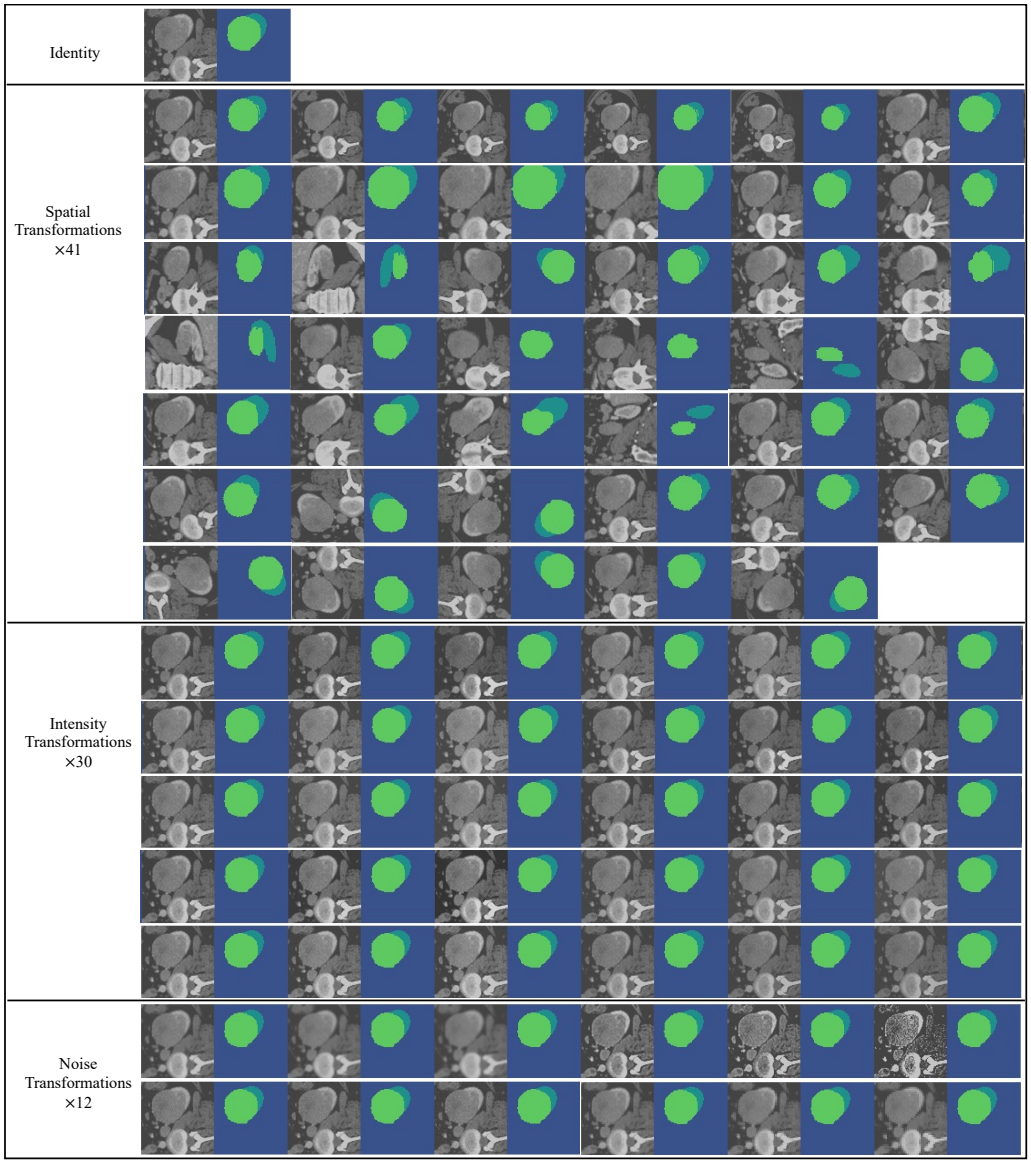


Figure 5.5: Visualization of K=84 choices of transformations for TEA we use in this study. We apply different transformations to a test sample of kidney tumor segmentation (KiTS).

5.3.11 Implementation Details

5.3.11.1 Derivatives calculation based on implicit function theorem

Similar to the descriptions in Section 4.4, in order to compute $\frac{\partial \theta^*}{\partial \hat{\boldsymbol{p}}_T}$, one can calculate the total derivatives on $\nabla_\theta \mathcal{L}_{train}(\theta^*, \hat{\boldsymbol{p}}_T) = \mathbf{0}$ w.r.t. $\hat{\boldsymbol{p}}_T$ from both sides, assuming that $\nabla_\theta \mathcal{L}_{train}(\theta^*, \hat{\boldsymbol{p}}_T)$ is continuously differentiable at $\mathbf{0}$ [15, 255]:

$$\frac{\partial \nabla_\theta \mathcal{L}_{train}(\theta^*, \hat{\boldsymbol{p}}_T)}{\partial \theta} \frac{\partial \theta^*}{\partial \hat{\boldsymbol{p}}_T} + \frac{\partial \nabla_\theta \mathcal{L}_{train}(\theta^*, \hat{\boldsymbol{p}}_T)}{\partial \hat{\boldsymbol{p}}_T} = \mathbf{0}. \tag{5.17}$$

Then, with the assumption that the Hessain $\nabla^2_\theta \mathcal{L}_{train}(\theta^*, \hat{\boldsymbol{p}}_T)$ is invertable, we can yield:

$$\frac{\partial \theta^*}{\partial \hat{\boldsymbol{p}}_T} = -(\nabla^2_\theta \mathcal{L}_{train}(\theta^*, \hat{\boldsymbol{p}}_T))^{-1} \nabla^2_{\theta, \hat{\boldsymbol{p}}_T} \mathcal{L}_{train}(\theta^*, \hat{\boldsymbol{p}}_T). \tag{5.18}$$

The results contain a Hessian $\nabla^2_{\theta, \hat{\boldsymbol{p}}_T} \mathcal{L}_{train}(\theta^*, \hat{\boldsymbol{p}}_T)$ and not practical to compute. Therefore, we follow the heuristics used in [73] to compute the derivatives.

5.3.11.2 Meta-gradient calculation

The gradient calculation in Eq. 5.8 can be further simplified based on finite difference approximation following [157], similar to what we have done in Section 4.4. With some small $\epsilon = 0.01 / \|\nabla_\theta \mathcal{L}_{val}(\theta^*)\|_2$, we calculate two new parameters with $\theta^\pm = \theta \pm \epsilon \nabla_\theta \mathcal{L}_{val}(\theta^*)$. With this notion, the second-order gradient can be written as:

$$\begin{aligned} \nabla^2_{\hat{\boldsymbol{p}}^t_T, \theta} \mathcal{L}_{train}(\theta^*, \hat{\boldsymbol{p}}^t_T) &\approx \frac{\nabla_{\hat{\boldsymbol{p}}^t_T} \mathcal{L}_{train}(\theta^+, \hat{\boldsymbol{p}}^t_T) - \nabla_{\hat{\boldsymbol{p}}^t_T} \mathcal{L}_{train}(\theta^-, \hat{\boldsymbol{p}}^t_T)}{2\epsilon \nabla_\theta \mathcal{L}_{val}(\theta^*)} \\ \Leftrightarrow \quad \nabla^2_{\hat{\boldsymbol{p}}^t_T, \theta} \mathcal{L}_{train}(\theta^*, \hat{\boldsymbol{p}}^t_T) \nabla_\theta \mathcal{L}_{val}(\theta^*) &\approx \frac{\nabla_{\hat{\boldsymbol{p}}^t_T} \mathcal{L}_{train}(\theta^+, \hat{\boldsymbol{p}}^t_T) - \nabla_{\hat{\boldsymbol{p}}^t_T} \mathcal{L}_{train}(\theta^-, \hat{\boldsymbol{p}}^t_T)}{2\epsilon}. \end{aligned} \tag{5.19}$$

In this way, we reduce the calculation complexity from $O(|\hat{\boldsymbol{p}}_T||\theta|)$ to $O(|\hat{\boldsymbol{p}}_T| + |\theta|)$ and can approximate Eq. 5.8 with two forward processes of f_{θ^*}.

5.3.11.3 Efficient data augmentation sampling

Sampling and applying different transformations to $\boldsymbol{x}$ (or $\tilde{\boldsymbol{x}}_i$) during each iteration is time-consuming. In practice, we bypass this bottleneck by fetching a number of transformed samples in advance. Then when we input the transformed samples into the model, we sample from the probability again and get the corresponding $\boldsymbol{s}_i$ as if it is sampled from the current distribution. In this way, the sam-

pling and data augmentation process can be done in parallel with network training and does not need additional time.

5.3.11.4 Efficient optimization

The proposed method described in Algorithm 2 would triple the training time compared with vanilla training process. However, the training time can be reduced by updating the data augmentation policies (step 8 and 9 in Algorithm 2) once several iterations. We find we can achieve similar results when the policies are updated once 10 iterations, only increasing the training time by 20%.

Table 5.2: List of all operations that our method can choose for TEA. There are 24 types of operations with a total of K=84 choices.

Category	Operation ID	Operation Name	Description	Range of magnitudes
Identity	0	Identity	No augmentation, $\mathcal{O}(\boldsymbol{x}) = \boldsymbol{x}$	None
Spatial transformations	1,2,3,4,5	Scaling down	Scale down the image by factor 1 + M	[0.05, 0.15, 0.25, 0.35, 0.45]
	6,7,8,9,10	Scaling up	Scale up the image by factor 1 + M	[0.05, 0.15, 0.25, 0.35, 0.45]
	11,12,13,14,15	RotateFrontal ACW	Rotate the image along frontal axis anticlockwise	[5°, 15°, 25°, 90°, 180°]
	16,17,18,19	RotateFrontal CCW	Rotate the image along frontal axis clockwise	[5°, 15°, 25°, 90°]
	20,21,22,23,24	RotateSagittal ACW	Rotate the image along sagittal axis anticlockwise	[5°, 15°, 25°, 90°, 180°]
	25,26,27,28	RotateSagittal CCW	Rotate the image along sagittal axis clockwise	[5°, 15°, 25°, 90°]
	29,30,31,32,33	RotateLongitudinal ACW	Rotate the image along longitudinal axis anticlockwise	[5°, 15°, 25°, 90°, 180°]
	34,35,36,37	RotateLongitudinal CCW	Rotate the image along longitudinal axis clockwise	[5°, 15°, 25°, 90°]
	38,39,40,41	Mirroring	Flip the sample in different planes	[Sagittal, Frontal, Axial, All]
Intensity transformations	42,43,44	Gamma expansion	Scale $\boldsymbol{x}$ to [0,1], then $\mathcal{O}(\boldsymbol{x}) = \boldsymbol{x}^{1+\gamma}$, and scale it back	[0.1, 0.3, 0.5]
	45,46,47	Gamma compression	Scale $\boldsymbol{x}$ to [0,1], then $\mathcal{O}(\boldsymbol{x}) = \boldsymbol{x}^{1/(1+\gamma)}$, and scale it back	[0.1, 0.3, 0.5]
	48,49,50	Inverted gamma expansion	Do the gamma compression with the inverted image	[0.1, 0.3, 0.5]
	51,52,53	Inverted gamma compression	Do the gamma expansion with the inverted image	[0.1, 0.3, 0.5]
	54,55,56	Adding intensity	$\mathcal{O}(\boldsymbol{x}) = \boldsymbol{x}$ + shift	[0.05, 0.15, 0.25]
	57,58,59	Subtracting intensity	$\mathcal{O}(\boldsymbol{x}) = \boldsymbol{x}$ - shift	[0.05, 0.15, 0.25]
	60,61,62	Scaling up intensity histogram	$\mathcal{O}(\boldsymbol{x}) = \boldsymbol{x}$ * (1 + scale)	[0.05, 0.15, 0.25]
	63,64,65	Scaling down intensity histogram	$\mathcal{O}(\boldsymbol{x}) = \boldsymbol{x}$ / (1 + scale)	[0.05, 0.15, 0.25]
	66,67,68	Increasing contrast	Reduce the image mean, then $\mathcal{O}(\boldsymbol{x}) = \boldsymbol{x}$ * (1 + scale), and add the mean value back	[0.05, 0.15, 0.25]
	69,70,71	Decreasing contrast	Reduce the image mean, then $\mathcal{O}(\boldsymbol{x}) = \boldsymbol{x}$ / (1 + scale), and add the mean value back	[0.05, 0.15, 0.25]
Noise transformations	72,73,74	Blurring	Blur the image using Gaussian filter with different standard deviations	[0.5, 0.7, 0.9]
	75,76,77	Sharpening	Sharpen the image using Laplacian of Gaussian filter with different standard deviations	[0.9, 0.7, 0.5]
	78,79,80	Adding Gaussian noise	Add Gaussian noise with different standard deviations	[0.025, 0.075, 0.125]
	81,82,83	Simulating low resolution	Scale down the image, and scale it back	[0.9, 0.7, 0.5]

5.4 Experiments

5.4.1 Experimental Setup

5.4.1.1 Network configurations

Our experiments are performed with DeepMedic [117] and a 3D U-Net [106]. We normalize all datasets using the pipeline of nnU-Net. Specifically, we adopt case-wise Z-score normalization for magnetic resonance (MR) images, and we normalize computerised tomography (CT) images with dataset-wise Z-score normalization based on foreground samples after clipping the Hounsfield units (HU) values from 0.5% to 99.5%. We choose cross entropy (CE) as $\mathcal{L}_{train}$ for DeepMedic and an equal combination of CE and soft Dice similarity coefficient (DSC) for 3D U-Net. We find that the optimal choice of $\mathcal{L}_{val}$ varies for different datasets. We adopt soft DSC for $\tilde{\mathcal{L}}_{val}$ for all the experiments. We choose the batch sizes n and m to be 10. We set the primary patch size as 37×37×37 for all the experiments with DeepMedic and a patch size of 64×64×64 for all the applications with 3D U-Net except prostate segmentation. We choose a patch size of 64×64×32 for prostate segmentation with 3D U-Net because images in this dataset have fewer slices. We train the networks for 1,000 epochs except for kidney and kidney tumor segmentation where we train for 2,000 epochs, as we observed that the networks need more iterations to converge on this task. All the reported results are the average of two runs with different random seeds.

5.4.1.2 Brain stroke lesion segmentation

We firstly evaluate the proposed method with binary brain stroke lesion segmentation using the dataset of anatomical tracings of lesions after stroke (ATLAS) [152]. The images have a voxel spacing of 1.0×1.0×1.0 mm. With a total of 220 T1-weighted MR images, we randomly select 73 (50%) or 145 (100%) for training, 31 for validation, and 44 for test. Note that this data split is different from what had in previous Chapters as previously we did not reserve data for validation but utilized data either for training or testing.

5.4.1.3 Kidney and kidney tumor segmentation

Secondly, we evaluate the proposed method with kidney and kidney tumor segmentation using the training dataset of KiTS challenge [96] which contains 210 CT images. We resample all images to voxel spacing of 1.6×1.6×3.2 mm. We randomly select 70 (50%) or 140 (100%) for training, 28 for validation, and 42 for test.

5.4.1.4 Abdominal organ segmentation

We also evaluate the proposed method with the task of abodominal organ segmentation which contains 14 classes including spleen (SP), right kidney (RK), left kidney (LK), gallbladder (GB), esopha-

gus (E), liver (LIV), stomach (STO), aorta (AO), inferior vena cava (IVC), portal vein and splenic vein (V), pancreas (PA), right adrenal gland (RA) and left adrenal gland (LA). We resample all images to a voxel spacing of $1.6\times1.6\times3.2$ mm. We randomly select 20 for training, 4 for validation, and 6 for test.

5.4.1.5 Cross-site prostate segmentation

Additionally, we utilize our method to align training data and validation data of prostate segmentation from two different sites A and B [158]. Specifically, we utilize 30 T2-weighted MR images from site A [21] which were collected with 1.5T Philips magnetic resonance imaging (MRI) machine with endorectal coil and 19 T2-weighted MR images from site B [137] which were collected with 3T Siemens MRI machines without endorectal coil. We resample all the images to a voxel spacing of $0.8\times0.8\times1.5$ mm. We investigate the scenario where the target domain (site B) has limited labeled data. We select 20 cases from site A for training, 4 cases for validation, and 6 cases for test. We select 1 case from site B for validation and use 18 cases for testing. Note that for cross-site prostate segmentation, we report results with models trained with both training data and validation data as this serves as a fairer baseline compared to using training data only.

5.4.2 Compared Methods

5.4.2.1 Heuristic

We compare with the heuristic TRA and TEA which are set as the default configurations in DeepMedic [117] and nnU-Net [106]. We also report a few results based on models trained using both the training and validation data with heuristic TRA.

5.4.2.2 Learned TRA

We compare with methods that adopt the data augmentation policies based on the validation performance without considering class dependency [52, 145, 153].

5.4.2.3 TRA with different transformation magnitudes

We also compare with RandAugment [54], which only changes the transformation magnitudes based on grid searching. Specifically, we keep the data augmentation probability and replace the operations of the same type with different magnitudes, yielding RandAugment-S, RandAugment-M and RandAugment-L.

We implement RandAugment [54] based on the heuristic TRA policies of nnU-Net [106]. Specifically, we keep the probability of adopting operations and substitute all the heuristic operations within the same type with the ones having certain magnitude. We create three RandAugment policies based on varied magnitudes, denoted as RandAugment-S, RandAugment-M and RandAugment-L. The

sampling distribution of these TRA policies are shown in Fig. 5.6. We choose small transformations for RandAugment-S, middle transformations for RandAugment-M and large transformations for RandAugment-L. In Fig. 5.6, we utilize darkness of different pie chart segments to represent the magnitudes of the operations. For example, the lightest grey segments refer to the operation without any transformations and the darkest (black) segments represent the operations with large transformations.

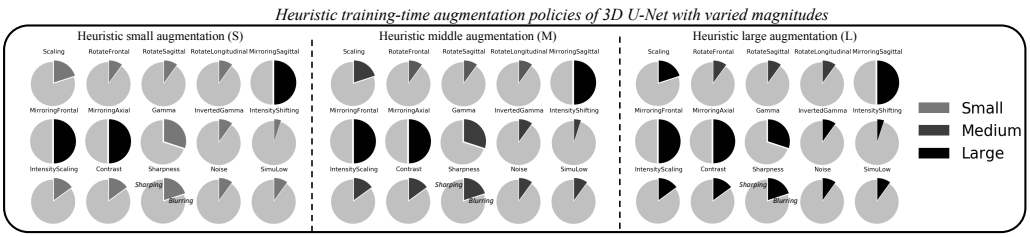


Figure 5.6: The sampling distribution for three heuristic TRA policies. The policies are are created by only changing the transformation magnitudes, referring to [54].

5.4.2.4 Learned TEA

We compare with methods which optimize TEA based on a pretrained segmentation model [124, 211]. Specifically, after training the model with proposed TRA, we refine TEA as described in Section 5.3.6.

5.4.3 Quantitative Results

Taking the manual segmentation as the ground truth, we calculate evaluation metrics including DSC, sensitivity (SEN), precision (PRC) and 95% Hausdorff distance (HD). The results are summarized in Table 5.3, 5.4, 5.5 and 5.6, separately. The learned probability distributions over augmentations for brain lesion segmentation based on different models with 100% ATLAS training data are summarized in Fig. 5.7.

5.4.3.1 The effectiveness of class-specific TRA

Heuristic TRAs, which were tuned based on varied segmentation tasks [106, 117], significantly help improve the segmentation performance in all cases compared with models trained without TRA. This indicates that TRA is vital for medical image segmentation as limited training data and class imbalance can easily lead to model overfitting [147].

Learned TRA, which is optimized with validation data, can provide application-specific policies and is more effective than heuristic TRA in most cases. We find that the models trained with learned TRA can even outperform the ones trained with heuristic TRA that use both training and validation

Table 5.3: Evaluation of brain stroke lesion segmentation on ATLAS based on different network architectures with different amounts of training data using different data augmentation methods. The best and second best results are in **bold** with the best also **underlined**.

Model	Training-time augmentation	Test-time augmentation	50% training data				100% training data			
			DSC	SEN	PRC	HD	DSC	SEN	PRC	HD
DeepMedic [117]	None	None	51.7	50.6	65.0	**20.4**	55.2	57.9	63.1	**24.3**
	Heuristic [117]	None	58.2	62.3	65.3	26.8	58.9	65.5	64.9	32.6
	Heuristic† [117]	None	**59.1**	64.7	65.0	30.3	**59.5**	63.0	67.5	29.3
	Learned [52, 145, 153]	None	**59.5**	62.7	68.5	**25.1**	58.4	62.9	67.0	26.2
	Learned Class-Specific	None	**59.5**	62.9	68.1	26.1	**60.2**	66.1	67.3	**22.7**
	Heuristic [117]	Heuristic [106]	60.1	63.7	68.7	**23.5**	60.6	65.7	67.7	27.6
	Learned Class-Specific	Heuristic [106]	**61.1**	64.4	70.2	25.6	**61.6**	66.2	68.6	**24.4**
	Learned Class-Specific	Learned [124, 211]	**61.3**	64.2	70.8	25.3	**61.6**	66.0	69.7	**23.5**
	Joint Learned Class-Specific		**61.3**	64.3	71.2	**24.5**	**61.9**	64.5	71.7	25.0
3D U-Net[47]	None	None	54.6	56.3	67.2	**32.6**	56.7	58.8	69.8	**23.0**
	Heuristic [106]	None	58.4	66.9	61.4	39.0	58.9	67.9	60.6	44.1
	Heuristic† [106]	None	58.8	67.8	59.8	52.2	58.3	69.5	56.2	58.6
	Learned [52, 145, 153]	None	**59.3**	66.6	61.1	40.9	**59.5**	69.2	61.1	48.2
	Learned Class-Specific	None	**62.0**	68.8	66.2	**37.8**	**62.1**	68.9	65.8	**34.8**
	Heuristic [106]	Heuristic [106]	61.7	67.0	69.6	22.0	62.3	68.6	68.4	31.4
	Learned Class-Specific	Heuristic [106]	61.8	66.4	70.2	**20.3**	**63.9**	68.4	72.2	**23.5**
	Learned Class-Specific	Learned [124, 211]	**62.2**	66.9	69.9	**20.5**	**63.9**	68.5	72.0	**24.5**
	Joint Learned Class-Specific		**62.3**	67.4	69.2	28.9	**64.0**	68.5	72.1	**24.5**

†We train these models with both training and validation data.

data, as shown in Table 5.3. This may indicate that it will be more effective to increase the heterogeneity within the training data by adopting application-specific TRA than adding a small amount of training data. We find RandAugment with specific magnitude could be more effective than the learned one in some cases. Specifically, RandAugment-L is better than the learned ones for kidney tumor segmentation under specific setting, as shown in Table 5.4. This might indicate that learned TRA is prone to overfitting the validation data and the optimized policies are not guaranteed to be optimal for unseen test data, as also found in [54].

In contrast, class-specific TRA can better model the heterogeneity of the real data by taking class imbalance into account, and thus overfit less and perform better on unseen test data than alternative methods. We argue that class-specific TRA is important as it concerns the imbalanced nature of the segmentation datasets and directly regularizes the training data in an implicit way. As shown in Fig. 5.7, compared with heuristic TRA, the learned policies tend to generate larger transformations for foreground samples while adopting smaller transformations to background samples. In segmentation, foreground classes are typically underrepresented and a learned baseline model would be biased towards the majority class. As a result, the model would map the foreground samples near the decision boundary and cause false negatives, as shown in [147]. Class-specific TRA can mitigate the class imbalance problem by inducing larger variance within the foreground samples, making the model learn a better decision boundary, consistently leading to better segmentation results with higher sensitivity. Particularly, we find class-specific TRA would improve

Table 5.4: Evaluation of kidney and kidney tumor segmentation based on different network architectures with different amounts of training data using different data augmentation methods. The best and second best results are in **bold** with the best also **underlined**.

Model	Training data	Training-time data augmentation	Test-time data augmentation	Kidney				Tumor			
				DSC	SEN	PRC	HD	DSC	SEN	PRC	HD
DeepMedic [117]	50%	None	None	92.6	89.7	97.0	**8.1**	40.1	35.4	56.2	93.0
		Heuristic [117]	None	95.5	95.1	96.4	16.2	66.6	69.3	72.4	76.3
		Learned [52, 145, 153]	None	**95.8**	95.4	96.4	11.7	**69.1**	71.3	75.1	**61.8**
		Learned Class-Specific	None	**95.7**	94.7	96.9	**9.8**	**71.6**	72.8	76.8	**66.5**
		Heuristic [117]	Heuristic [106]	**95.8**	95.0	97.1	11.0	70.5	70.5	78.5	58.7
		Learned Class-Specific	Heuristic [106]	**95.7**	94.9	97.0	**6.0**	72.5	73.4	78.3	**48.1**
		Learned Class-Specific	Learned [124, 211]	**95.8**	94.9	97.1	**5.9**	**72.8**	73.6	78.5	**47.9**
		Joint Learned Class-Specific		**95.8**	94.9	97.0	11.0	**73.3**	73.5	79.7	48.4
	100%	None	None	94.7	93.5	96.7	**10.6**	51.1	50.0	62.0	**72.8**
		Heuristic [117]	None	**96.0**	96.8	95.3	**18.0**	**69.5**	77.2	69.8	**76.3**
		Learned [52, 145, 153]	None	95.0	97.2	93.2	23.5	**69.5**	79.3	67.6	89.4
		Learned Class-Specific	None	**95.8**	97.1	94.8	19.4	**71.2**	78.1	70.4	88.3
		Heuristic [117]	Heuristic [106]	**96.3**	96.8	96.0	**13.3**	72.9	77.9	74.1	62.5
		Learned Class-Specific	Heuristic [106]	96.0	97.1	95.1	22.3	73.1	79.5	73.2	**57.2**
		Learned Class-Specific	Learned [124, 211]	**96.1**	97.1	95.2	21.9	**73.3**	79.3	73.6	**60.5**
		Joint Learned Class-Specific		**96.1**	96.8	95.5	**12.7**	**74.1**	79.6	74.0	71.7
3D U-Net [47]	50%	None	None	95.3	94.0	97.3	5.7	43.5	39.5	60.9	104.6
		Heuristic [106]	None	**96.6**	96.4	96.9	**2.6**	76.6	80.2	77.4	**40.6**
		RandAugment-S [54]	None	96.4	96.0	97.0	2.7	74.4	76.6	78.3	45.5
		RandAugment-M [54]	None	96.4	95.7	97.2	2.8	77.5	79.0	79.7	**34.2**
		RandAugment-L [54]	None	96.4	96.0	97.0	2.7	**77.6**	82.1	77.0	61.4
		Learned [52, 145, 153]	None	96.5	96.3	96.8	2.8	76.7	82.0	76.1	55.7
		Learned Class-Specific	None	**96.8**	96.6	96.9	**2.5**	**78.4**	82.2	78.0	47.2
		Heuristic [106]	Heuristic [106]	**96.9**	96.6	97.2	**2.3**	**78.8**	82.1	79.4	**37.2**
		Learned Class-Specific	Heuristic [106]	96.8	96.5	97.1	2.5	78.7	81.7	79.6	**42.0**
		Learned Class-Specific	Learned [124, 211]	96.8	96.5	97.1	2.5	**78.8**	81.7	79.6	**42.0**
		Joint Learned Class-Specific		**97.0**	96.9	97.2	**2.2**	**79.3**	82.2	79.7	45.4
	100%	None	None	96.4	95.8	97.4	2.8	60.3	57.1	71.6	78.4
		Heuristic [106]	None	**96.9**	97.0	96.9	**2.3**	**77.6**	82.1	77.6	59.5
		Learned [52, 145, 153]	None	**96.8**	97.0	96.8	**2.4**	**78.5**	84.2	77.4	**50.6**
		Learned Class-Specific	None	**96.8**	96.8	96.8	2.5	**78.5**	83.2	78.0	**48.8**
		Heuristic [106]	Heuristic [106]	**97.0**	97.0	97.1	**2.2**	79.7	83.3	79.5	45.5
		Learned Class-Specific	Heuristic [106]	**97.0**	97.0	97.0	2.4	**80.5**	84.0	80.2	42.3
		Learned Class-Specific	Learned [124, 211]	**97.0**	97.0	97.0	2.4	**80.6**	84.1	80.4	**38.3**
		Joint Learned Class-Specific		**97.1**	97.0	97.2	**2.1**	80.4	83.5	80.6	**38.6**

the segmentation performance of rare classes more significantly (c.f. Table 5.5) as it can enhance the rare class representation by increasing the heterogeneity of foreground sample variation. We also find that the probabilities of spatial transformations change more significantly compared to intensity transformations. This might indicate that spatial transformations are more effective in increasing the heterogeneity within training data. We report the segmentation results of both site B and site A with cross-site prostate segmentation in Table 5.6. Although the learned data augmentation is optimized based on the validation data from site B (target domain), the models can still generalize well on site A. This indicates that our method can be a safe choice to calibrate the segmentation performance of different domains within multi-domain learning.

We find that class-specific TRA could be less effective with 3D U-Net on prostate segmentation. It is mainly because the sampled patches always contain foreground, as the image size of this dataset

Table 5.5: Evaluation of abdominal organ segmentation based on different network architectures using different data augmentation methods. The best and second best results are in **bold** with the best also **underlined**. AVG_r is the average performance of all rare classes including GB, E, AO, IVC, V, PA, RA and LA.

Model	Training-time data augmentation	Test-time data augmentation	DSC														
			SP	RK	LK	GB	E	LIV	STO	AO	IVC	V	PA	RA	LA	AVG	AVG_r
DeepMedic [117]	None	None	87.7	89.5	91.6	48.7	71.9	94.8	76.7	83.8	82.5	62.0	51.9	49.4	54.3	72.7	63.1
	Heuristic [117]	None	90.9	90.5	90.8	59.8	75.1	93.0	77.9	84.3	84.9	69.8	65.1	65.4	66.4	**78.0**	71.3
	Learned [52, 145, 153]	None	93.4	90.8	93.1	63.6	76.0	94.1	81.7	86.2	84.2	71.5	66.7	68.2	64.6	**79.5**	**72.6**
	Learned Class-Specific	None	91.8	89.7	91.0	65.7	76.6	93.9	79.6	85.2	85.2	69.9	68.9	69.7	66.8	**79.5**	**73.5**
	Heuristic [117]	Heuristic† [106]	91.4	90.6	92.2	60.3	78.6	93.1	78.1	84.4	85.3	70.5	66.4	65.4	69.2	78.9	72.5
	Learned Class-Specific	Heuristic† [106]	92.1	90.2	92.1	65.3	77.3	94.0	80.8	86.6	86.1	69.2	68.5	66.7	64.9	79.5	**73.1**
	Learned Class-Specific	Learned [124, 211]	92.1	89.9	91.5	65.6	77.5	94.0	80.4	86.3	86.0	69.7	68.8	68.7	66.6	**79.8**	**73.7**
	Joint Learned Class-Specific		92.9	91.5	93.3	63.7	77.6	93.4	80.0	86.1	85.1	69.9	68.5	66.3	66.8	**79.6**	73.0
3D U-Net [47]	None	None	89.2	91.6	92.5	29.9	69.8	95.2	86.9	88.1	86.1	69.3	60.0	54.8	61.7	75.0	65.0
	Heuristic [106]	None	94.9	92.7	92.6	55.4	75.9	96.0	89.3	91.6	88.0	72.9	75.4	67.7	67.8	**81.5**	74.3
	Learned [52, 145, 153]	None	94.6	92.9	92.8	59.7	76.6	96.1	90.0	91.8	88.2	73.1	73.3	64.4	66.0	**81.5**	**74.1**
	Learned Class-Specific	None	95.0	93.4	92.9	59.1	76.0	96.2	88.9	91.9	88.0	73.8	74.3	61.9	72.6	**81.8**	**74.7**
	Heuristic [106]	Heuristic [106]	95.3	93.6	92.8	58.4	73.2	96.2	90.4	91.8	88.6	74.6	75.2	64.5	69.0	81.8	74.4
	Learned Class-Specific	Heuristic [106]	95.3	93.4	93.0	52.7	77.5	96.1	89.9	92.0	88.2	76.2	74.7	64.8	72.8	82.0	74.9
	Learned Class-Specific	Learned [124, 211]	95.3	93.3	92.9	56.1	77.6	96.2	89.9	92.0	88.3	75.7	74.7	64.8	72.6	**82.2**	**75.2**
	Joint Learned Class-Specific		94.4	93.5	93.1	61.8	80.8	96.0	89.3	90.9	87.7	72.7	77.6	66.7	71.4	**82.8**	**76.2**

†We adopt a heuristic TEA policy with larger probability of identity transformation here because typical ones would decrease the performance.

is relatively small. In this case, the optimization could be misled by the class-specific constraints. In practice, this could be resolved by adopting a smaller patch size or restricting the regions of loss calculation.

5.4.3.2 The effectiveness of joint optimization

We find heuristic TEA can help the pretrained models produce better overall segmentation results with higher precision. This is because the ensemble of multiple predictions can reduce false positives as the models are unlikely to produce the same kind of false positives with all the transformed images. However, when TRA is optimized based on validation data without TEA , heuristic TEA might not work well as the model may overfit to the original data distribution and thus fail to generalize to the transformed data. Specifically, we observe that heuristic TEA would decrease the model performance for 3D U-Net trained with 50% ATLAS training data (-0.3 in terms of DSC, c.f. Table 5.3) and Deepmedic trained for prostate segmentation (-0.1 in terms of DSC, c.f. Table 5.6) using learned class-specific TRA. In comparison, learned TEA can refine the transformations to fit the pretrained models and improve the results for most cases.

However, learned TEA alone does not affect model training and cannot change the results significantly compared to heuristic TEA . In contrast, our method optimizes TRA based on TEA along the training process, jointly aligning the data distributions resulting in larger overlaps. For example, as illustrated in Fig. 5.7(a), the learned TEA policy would increase the probability of flipping in sagittal planes for DeepMedic trained with 100% ATLAS training data. Correspondingly, TRA would tune the training data distribution with more samples flipped in the sagittal planes. It might be because the left and right hemispheres of human brains are generally symmetric. In this way, the segmentation

Table 5.6: Evaluation of cross-site prostate segmentation based on different network architectures using different data augmentation methods. The best and second best results are in **bold** with the best also **underlined**.

Model	Site A Training-time data augmentation	Site B Test-time data augmentation	Site B DSC	Site B SEN	Site B PRC	Site B HD	Site A DSC	Site A SEN	Site A PRC	Site A HD
DeepMedic [117]	None	None	14.9	11.6	45.3	42.6	82.4	77.1	90.7	6.7
	Heuristic [117]	None	46.4	43.2	59.4	26.9	88.0	85.5	91.4	4.8
	Heuristic† [117]	None	56.7	46.4	77.5	**9.4**	27.6	20.0	81.4	18.9
	Heuristic‡ [117]	None	**69.3**	67.2	73.5	**15.1**	**88.1**	84.8	92.3	**4.5**
	Learned‡ [52, 145, 153]	None	65.8	62.8	75.1	21.9	87.5	83.4	92.6	**4.6**
	Learned Class-Specific‡	None	**70.0**	68.0	75.9	18.7	**88.2**	85.6	91.5	4.7
	Heuristic‡ [117]	Heuristic [106]	69.4	66.3	76.5	**8.0**	88.2	84.6	92.6	4.6
	Learned Class-Specific‡	Heuristic [106]	69.9	66.3	80.0	**8.0**	**88.8**	86.1	92.4	**4.4**
	Learned Class-Specific‡	Learned [124, 211]	**70.2**	67.7	77.6	15.3	**88.5**	86.0	91.7	4.6
	Joint Learned Class-Specific‡		**72.8**	71.0	76.6	**7.9**	88.2	85.4	91.8	**4.5**
3D U-Net [47]	None	None	57.2	52.8	69.7	**13.7**	87.1	84.2	91.1	**5.3**
	Heuristic [106]	None	63.3	89.3	50.9	64.0	**89.4**	88.4	90.8	16.0
	Heuristic† [106]	None	68.7	63.9	84.1	**9.8**	55.6	48.6	77.2	25.1
	Heuristic‡ [106]	None	**73.9**	88.4	65.6	60.1	**89.4**	88.9	90.0	18.7
	Learned‡ [52, 145, 153]	None	**76.5**	89.8	67.9	44.2	87.0	89.2	85.3	26.9
	Learned Class-Specific‡	None	73.2	90.1	63.4	41.5	**87.7**	89.6	86.6	**15.4**
	Heuristic‡ [106]	Heuristic [106]	78.7	90.5	71.4	28.0	**89.7**	89.1	90.6	**12.1**
	Learned‡ [52, 145, 153]	Heuristic [106]	78.5	89.2	71.6	31.1	88.9	89.2	89.0	12.8
	Learned‡ [52, 145, 153]	Learned [124, 211]	**79.6**	89.9	72.7	**23.6**	89.0	89.3	89.1	12.3
	Joint Learned‡		**80.0**	88.2	74.1	**18.9**	**90.0**	89.3	91.0	**11.8**

†We pretrain these models with training data from site A and fine-tune with validation data from site B.

‡We train these models with both training data from site A and validation data from site B.

models not only have lower risks of making the same false positives but also generalize better on varied transformed samples. As a result, we find that the joint optimization further boosts the segmentation performance by achieving higher precision and sensitivity.

We argue that the joint optimization is crucial for data augmentation as it explicitly aligns the training and test-time conditions. Otherwise, the model may get stuck into a local minimum where we cannot find effective test-time transformations to fit the training data distribution. For example, we find that when compared with segmentation without TEA, learned TEA brings limited improvements for 3D U-Net trained with 50% KiTS training data (0.4 in terms of DSC, c.f. Table 5.4) and DeepMedic trained for prostate segmentation (0.2 for site B in terms of DSC, c.f. Table 5.6). This indicates that the predictions on most chosen transformations cannot contribute much to the results on top of the predictions of the original test images. In contrast, the joint optimization leverages the varied test-time transformations and improve the segmentation (0.9 and 2.8 separately in terms of DSC). We notice that the learned TEA policies would generally prefer the original images (identity). In addition, the transformations which are not included in heuristic TEA are hardly useful. These findings indicate that we may not need to apply large transformations to the test data to improve generalization.

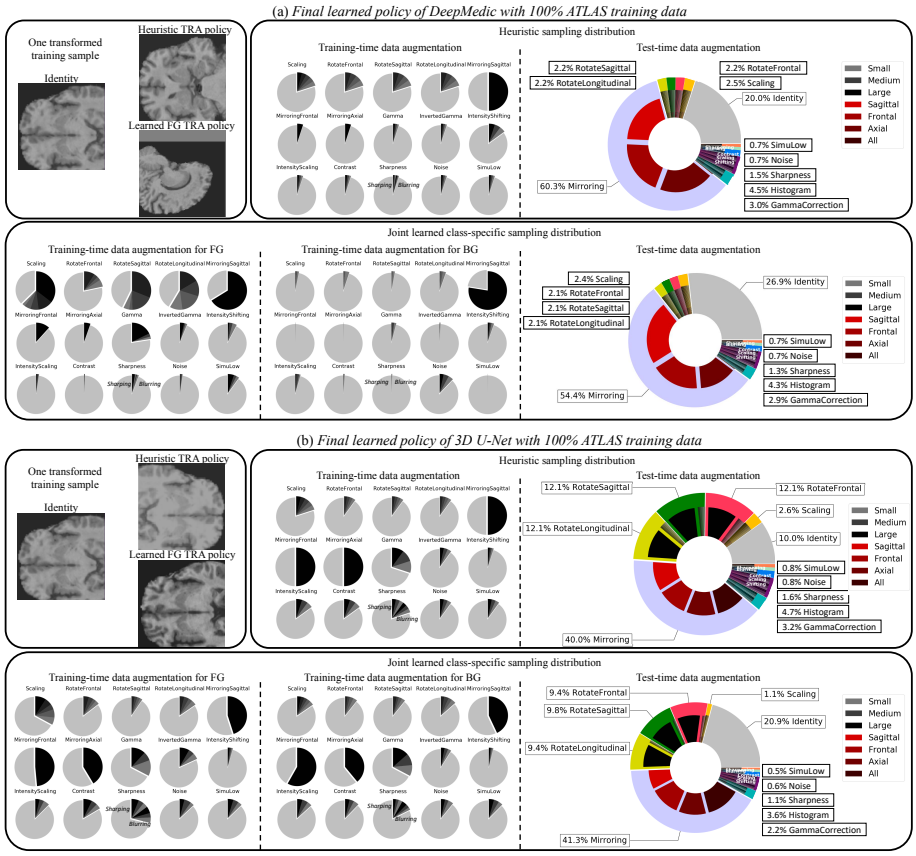


Figure 5.7: The heuristic data augmentation policy and the learned probability distributions over augmentations based on different segmentation models for brain stroke lesion segmentation with 100% ATLAS training data. We also visualize an example of the transformed foreground training sample with different sampling distributions for TRA. Our framework provides application-specific and class-specific data augmentation policies. We find the learned policies would adopt larger transformations to the foreground (FG) than the background (BG) samples, implicitly alleviating the class imbalance issue.

We visualize some segmentation results in Fig. 5.8. Similar to the findings in a previous study [147], the model trained with imbalanced dataset would be prone to undersegment the foreground samples as a result of overfitting under class imbalance. Our class-specific TRA model can significantly reduce false negatives and improve the sensitivity of segmentation results. We observe heuristic TEA could cause under-segmentation while the joint optimization can further help the model improve segmentation performance by identifying more foreground samples.

5.4.4 Learned Policies with Different Settings

We summarize all the learned policies for different tasks based on different models. We expect it to be taken as a reference for participants to collect datasets and design transformations for data

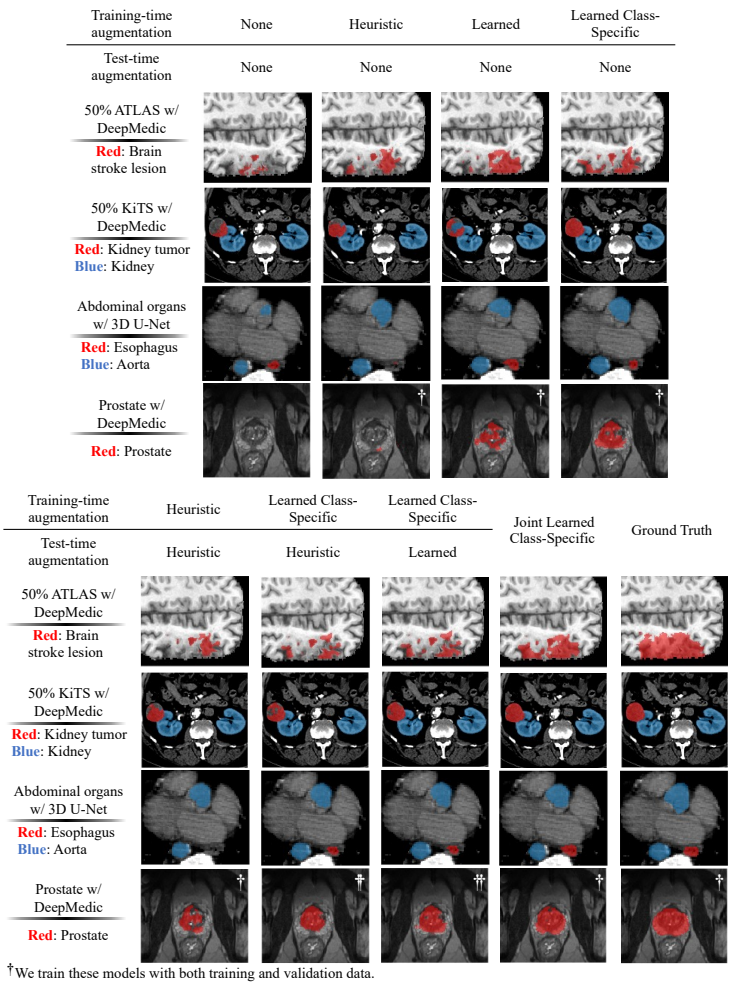


Figure 5.8: Visualization of different datasets and segmentation results with different data augmentation methods. The proposed data augmentation framework can help the model produce overall better segmentation results with higher sensitivity. Best viewed in color.

augmentation. The data augmentation policies for different network architectures are initialized with the heuristic policies provided by these frameworks. These default policies are designed for general purpose and specifically to fit the properties of different segmentation models. Therefore, the learned policies also differ considerably between models.

We summarize the learned data augmentation policies for brain stroke lesion segmentation with 50% ATLAS training data in Fig. 5.9, As brain stroke lesion is relatively small and always under-represented, we find the learned TRA policies tend to apply transformations with larger magnitude to foreground samples and transformations with smaller magnitude to background samples. Specifically, when compared with heuristic policies, the learned policies would increase the probabilities of adopting transformations for foreground samples while decrease these probabilities for

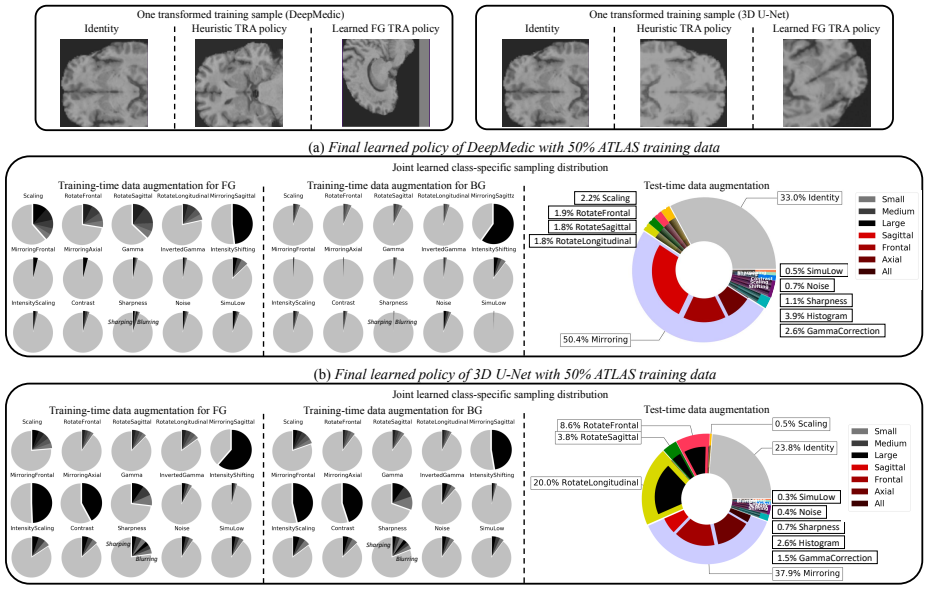


Figure 5.9: The learned probability distributions over augmentations based on different segmentation models for brain stroke lesion segmentation with 50% ATLAS training data. We also visualize an example of the transformed foreground training sample with different sampling distributions for TRA.

background samples. We observe consistent changes in all kinds of transformations. This indicates that the segmentation models would benefit from foreground samples with more variances and background samples with limited transformations.

We find the probabilities of spatial transformations such as scaling and rotation would be largely increased. This indicates that spatial transformations make more differences to the training data distributions. We also find that the policies learned with 100% training data would often adopt larger transformations when compared with the ones learned with 50% training data. This indicates that the optimal TRA policies vary for training datasets with different amounts of samples. Specifically, we should utilize larger transformations to effectively extend the data distributions with sufficient training data. It may be because small transformations are hard to add more information to the training datasets on the top of sufficient training data.

We find the learned TEA policies for DeepMedic often increase the probabilities of flipping in sagittal planes. It might be because the initialized TRA policy for DeepMedic has large probabilities of flipping in the sagittal planes (which is designed by taking into account the symmetrical brain structure). We also notice the learned TEA policies would always largely increase the probabilities of identity for both DeepMedic and 3D U-Net. This indicates that the predictions of images in the original data distribution are fairly accurate.

We summarize the learned data augmentation policies for kidney and kidney tumor segmentation

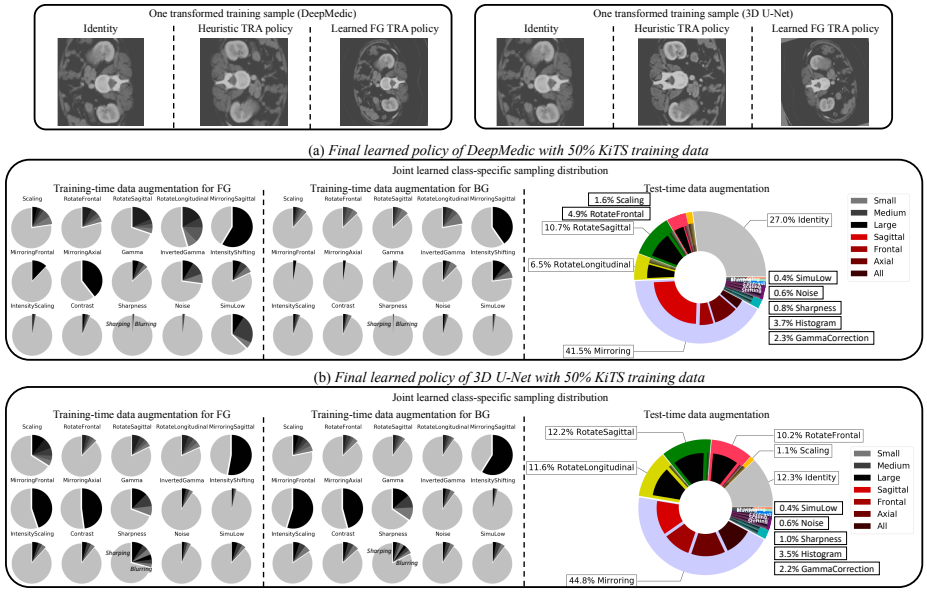


Figure 5.10: The learned probability distributions over augmentations based on different segmentation models for kidney and kidney tumor segmentation with 50% KiTS training data. We also visualize an example of the transformed foreground training sample with different sampling distributions for TRA.

with 50% KiTS training data in Fig. 5.10 and 100% KiTS training data in Fig. 5.11. We find when the segmentation models are trained with 50% training data, the learned TRA policies tend to generate larger transformations for foreground samples, similar to the case of brain stoke lesion segmentation. This might be because the kidney and kidney tumor are under-represented with less training data. In contrast, the policies learned with 100% training data do not have consistent bias towards the foreground samples. This may be because the class imbalance problem probably would not affect the learning process too much as the training datasets contain a sufficient amount of foreground samples. Under such condition, the policies are learned to generate class-specific transformations. Specifically, the learned policies increase the probabilities of scaling for foreground samples while increase the probabilities of noise transformations such as sharpening and simulating low resolution for background samples.

When compared with brain stroke lesion segmentation in T1-weighted MR images, we find the learned policies prone to adopting intensity transformations such as gamma correction and intensity shifting and noise transformations such as adding Gaussian noise and simulating low resolution for TRA for kidney and kidney tumor segmentation in CT images. This is may because the objects in CT images have low contrast and blurred boundaries. In this case, the simulated images with varied imaging quality can help the segmentation model generalize better with unseen image conditions.

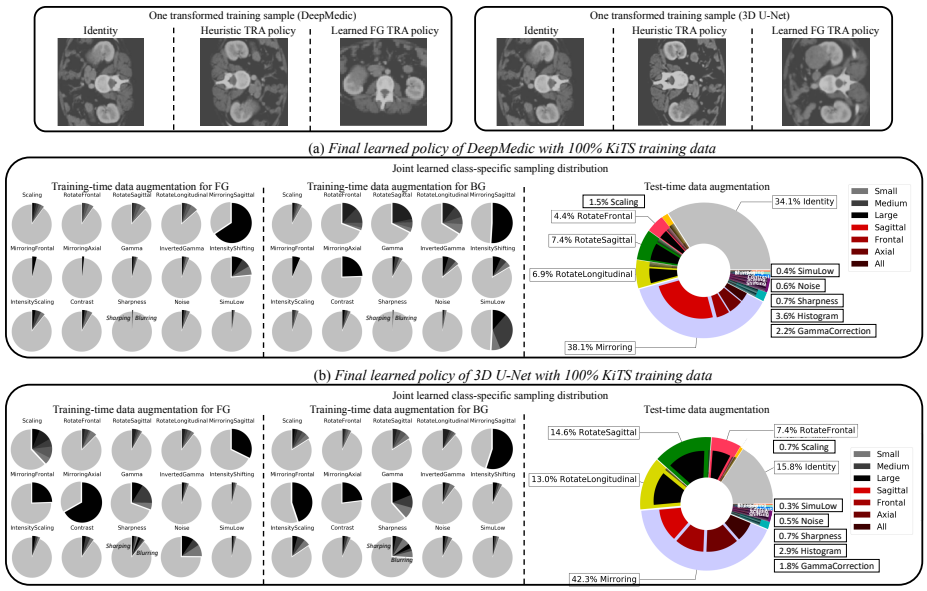


Figure 5.11: The learned probability distributions over augmentations based on different segmentation models for kidney and kidney tumor segmentation with 100% KiTS training data. We also visualize an example of the transformed foreground training sample with different sampling distributions for TRA.

We summarize the learned data augmentation policies for abdominal organ segmentation in Fig. 5.12. The learned policies for abdominal organ segmentation are quite different from the cases of brain stroke lesion and kidney tumor segmentation. This is may due to the complexity of foreground class which contains many different classes of abdominal organs. We find the learned TRA policies are prone to adopt large scaling transformations to the background samples. This maybe because some background objects which are similar to the foreground objects vary in size. The segmentation models can perform better when learned simulated background objects with varied scales. Similar to the case of kidney tumor segmentation in CT images, the learned TRA policies tend to adopt many intensity and noise transformations. We think this is also related to the low imaging contrast and blurry boundaries of CT.

We find the default TEA policy would decrease the performance of the segmentation model, therefore we utilize and initialize with a TEA policy with increased probability of identity, as shown in the upper part of Fig. 5.12. The learned TEA policy for DeepMedic further increases the probability of identity. This indicates that DeepMedic cannot perform well with transformed images. This is because the initialized TRA policies for DeepMedic do not contain large transformations. This could be also related to the network architecture of DeepMedic which contains more convolutional layers in the original size. The network design drives DeepMedic to make predictions relying more on the local features, potentially being more sensitive to noise.

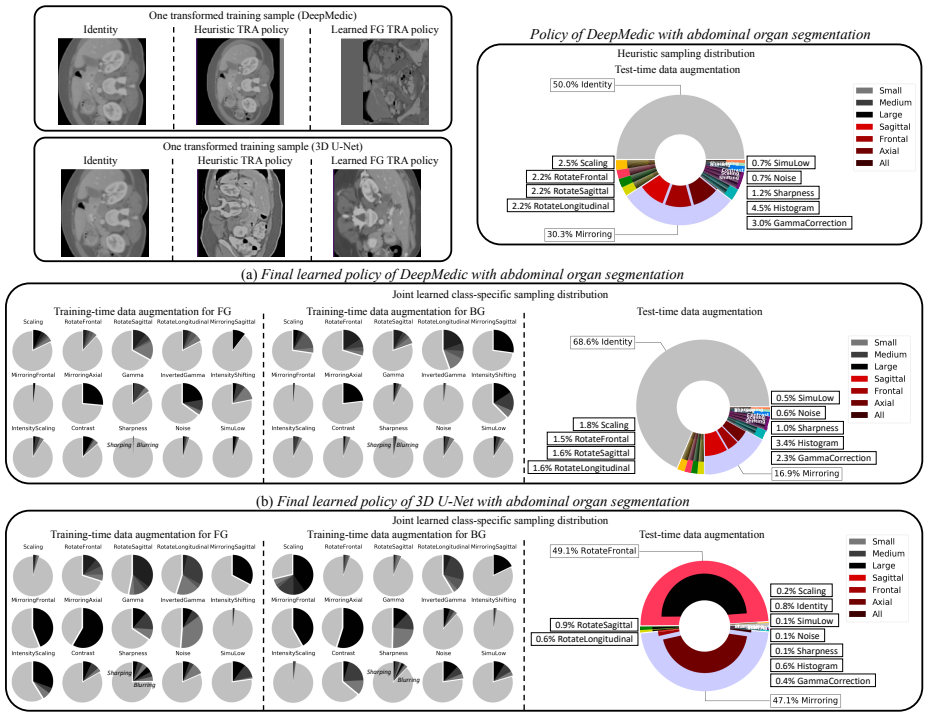


Figure 5.12: The learned probability distributions over augmentations based on different segmentation models for abdominal organ segmentation. We also visualize an example of the transformed foreground training sample with different sampling distributions for TRA.

We summarize the learned data augmentation policies for cross-site prostate segmentation in Fig. 5.13. We find the learned TRA policies are generally very close to the initialized ones. This indicates that the default TRA policies fit this task well. We notice the learned TRA policies for background adopt more intensity transformations such as intensity shifting and noise transformations such as simulating low resolution which downsamples the image and scale it back. Those transformations might help align the MRI datasets acquired with different settings.

5.4.5 Logit Distribution Maps

To obtain a better understanding of network behaviour after training with different data augmentation methods, we monitor the network activations when processing test samples of different classes. The logit distributions of DeepMedic trained with 100% ATLAS training data and 50% KiTS training data are both summarized in Fig. 5.14. We find that data augmentation can always help the model build better representation, especially for the minority classes such as lesion and kidney tumor. Without data augmentation, the model would map the samples from the minority class across the decision boundary, causing false negatives. We observe that heuristic data augmentation can help the model map the logits of the samples from the minority classes away from the decision boundary. The proposed data augmentation strategy can help the model further reduce logit shifts

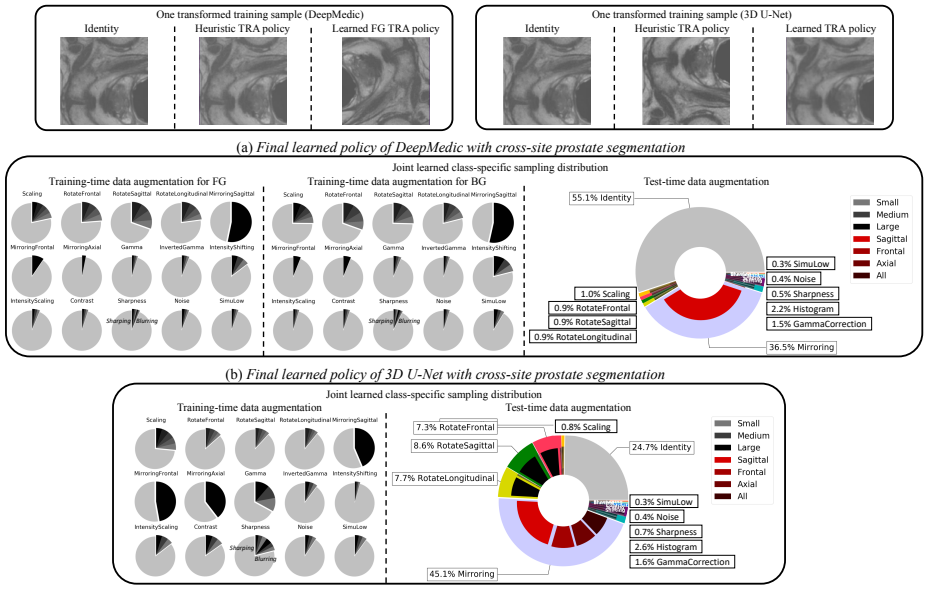


Figure 5.13: The learned probability distributions over augmentations based on different segmentation models for cross-site prostate segmentation. We also visualize an example of the transformed foreground training sample with different sampling distributions for TRA.

of the minority classes and build a better decision boundary. Moreover, it can make the model map the logits of samples from the same class to a more compact cluster. This indicates that the model trained with our method can implicitly encourage the feature to have better inter-class separability and intra-class compactness, thus generalize well.

5.4.6 Visualization of Validation Loss

We visualize validation loss curves of four settings in Fig. 5.15 and Fig. 5.16. We find when the model is trained without TRA, it would be very likely to overfit the training dataset as we observe that the calculated CE of validation data would even increase during the training process, but not for DSC. This might indicate that the model becomes overconfident with the prediction of the hard cases.

Heuristic and learned TRA can help the model decrease the validation loss, while the proposed learned class-specific TRA is the most effective to decrease the validation loss. This indicates that our class-specific transformation model is more capable of mimicking the underlying data distribution and thus help the segmentation model generalize better.

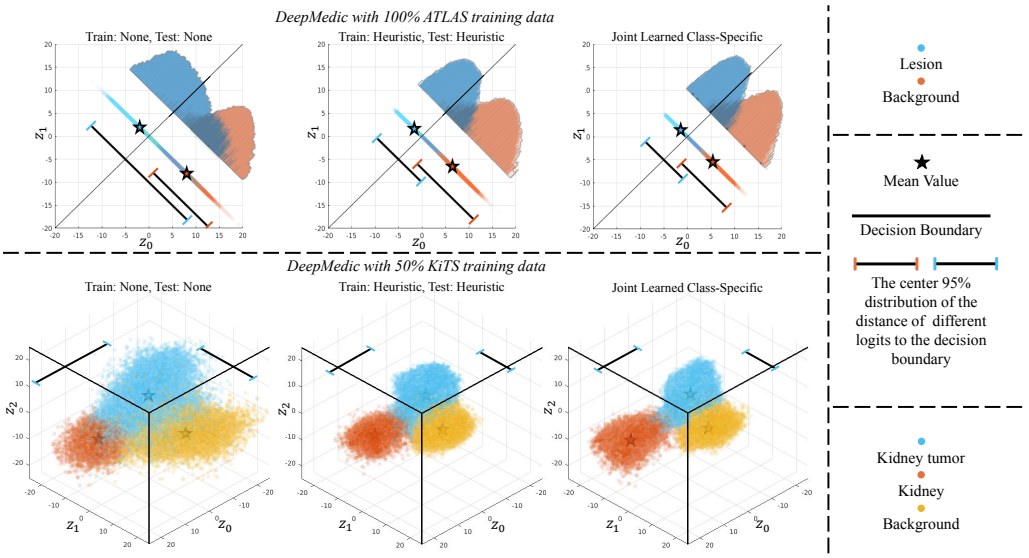


Figure 5.14: The logit distribution learned with different data augmentation policies. (Uppr part) Activations of the classification layer (logit z_0 for backround and logit z_1 for lesion) when processing the lesion (blue) or background samples (orange) of test data with different data augmentation methods. (Lower part) Activations of the classification layer (logit z_0 for background, logit z_1 for kidney and logit z_2 for kidney tumor) when processing the kidney tumor (blue), kidney (yellow) and background samples (orange) of test data with different data augmentation methods. Compared with training without data augmentation or heuristic data augmentation, the proposed framework can make the model better separate the samples from different classes.

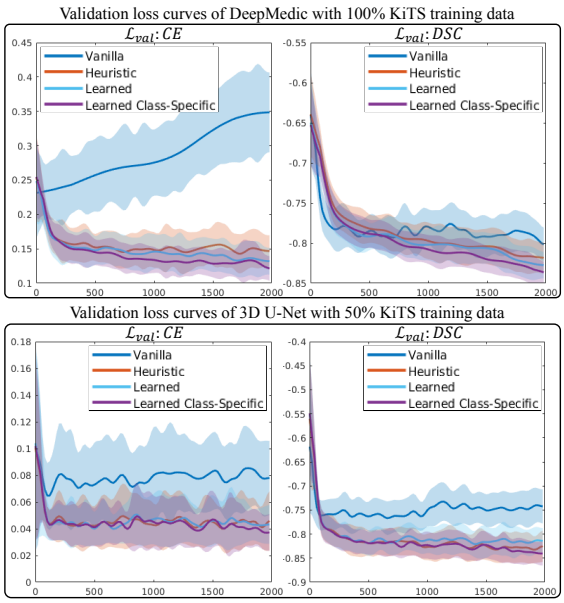


Figure 5.15: Validation loss curves of different models trained for kidney and kidney tumor segmentation with KiTS. Compared with heuristic and learned class-agnostic data augmentation, the learned class-specific data augmentation can decrease the loss more. This may indicate that class-specific data augmentation is more effective in aligning the training and validation data distribution.

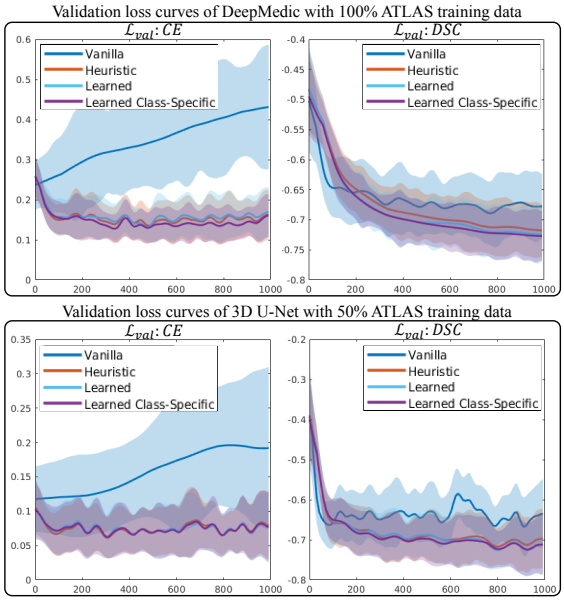


Figure 5.16: Validation loss curves of different models trained for brain lesion segmentation with ATLAS. The learned class-specific data augmentation can decrease the loss more in comparison with the alternative methods.

5.5 Conclusion

We presented a general data augmentation framework for medical image segmentation. Compared with current solutions, our method aims to bridge the gap between training and test data distributions by class-specific TRA and joint optimization of TRA and TEA. We observe promising improvements in various tasks and models, making the proposed framework an attractive alternative to heuristic data augmentation strategies.

The proposed data augmentation framework determines the sample class according to the category of the center pixel of the patch. We think class-specific TRA fits the task of image classification well and should perform well with long-tailed classification tasks. However, this could be not compatible with segmentation tasks when training with large patch size, as in this case each patch would contain all classes of samples. In the future, we could improve our method to overcome this limitation by restricting the regions for loss calculation. Specifically, we can apply varied class-specific TRA to the same patch and calculate the cost functions only with pixels from specific classes.

Our data augmentation framework does not include any new kinds of data augmentation but search the optimal ones within the predefined searching pool. The learned data augmentation could not bring much benefits when the initialized policies already fit the given task well. We can extend the transformation sets for both TRA and TEA with more advanced data augmentation strategies for specific tasks to achieve better performance.

We summarize all the learned policies of our experiments with varied tasks and models. We believe that the learned policies can provide valuable insights for practitioners to inform dynamic data collection and future designs of image transformations for data augmentation.

Chapter 6

Estimating Performance under Domain Shifts with Class-Specific Confidence Scores

Motivated by the observation of performance degradation when a model is deployed with data from different sites in Chapter 5, in this chapter we look into the safe deployment of pretrained model under domain shifts. Similar to class-specific modifications in Chapter 5, we adopt class-specific parameters for different classes, aiming at improving model performance accuracy. The work in this chapter is based on the publication:

1. **Z. Li**, K. Kamnitsas, M. Islam, C. Chen and B. Glocker. Estimating Model Performance under Domain Shifts with Class-Specific Confidence Scores. International Conference on Medical Image Computing and Computer-Assisted Intervention, 2022: 693-703.

Machine learning models could fail to perform well in an unexpected way because of domain shifts. If we could estimate the performance that a pre-trained model would achieve on data from a specific deployment setting, for example a certain clinic, we could judge whether the model could safely be deployed or if its performance degrades unacceptably on the specific data. Existing approaches estimate this based on the confidence of predictions made on unlabeled test data from the deployment's domain. We find that existing methods struggle with data that present class imbalance, because the methods used to calibrate confidence do not account for bias induced by class imbalance, consequently failing to estimate class-wise accuracy. We introduce class-wise calibration within the framework of performance estimation for imbalanced datasets. Specifically, we derive class-specific modifications of state-of-the-art (SOTA) confidence-based model evaluation methods including temperature scaling (TS), difference of confidences (DoC), and average thresholded confidence (ATC). We also extend the methods to estimate Dice similarity coefficient (DSC) in image segmentation. We conduct experiments on four tasks and find the proposed modifications consistently improve the estimation accuracy for imbalanced datasets. Our methods reduce the estimation error by 18% in classification under natural domain shifts, and reduce the estimation accuracy by 50% on segmentation tasks, when compared with prior methods.

6.1 Introduction

Based on the independently and identically distributed (IID) assumption, one can estimate the model performance with labeled validation data which is collected from the same source domain as the training data. However, in many real-world scenarios where test data is typically sampled from a different distribution, the IID assumption does not hold anymore and the model's performance tends to degrade because of domain shifts [14]. It is difficult for practitioner to assess the model reliability on the target domain as labeled data for a new test domain is usually not available. It would be of great practical value to develop a tool to estimate the performance of a trained model on an unseen test domain without access to ground truth [67]. For example, for a skin lesion classification model which is trained with training data from the source domain, we can decide whether to deploy it to a target domain based on the predicted accuracy of a few unlabeled test images. In this study we aim to estimate the model performance by only making use of unlabeled test data from the target domain.

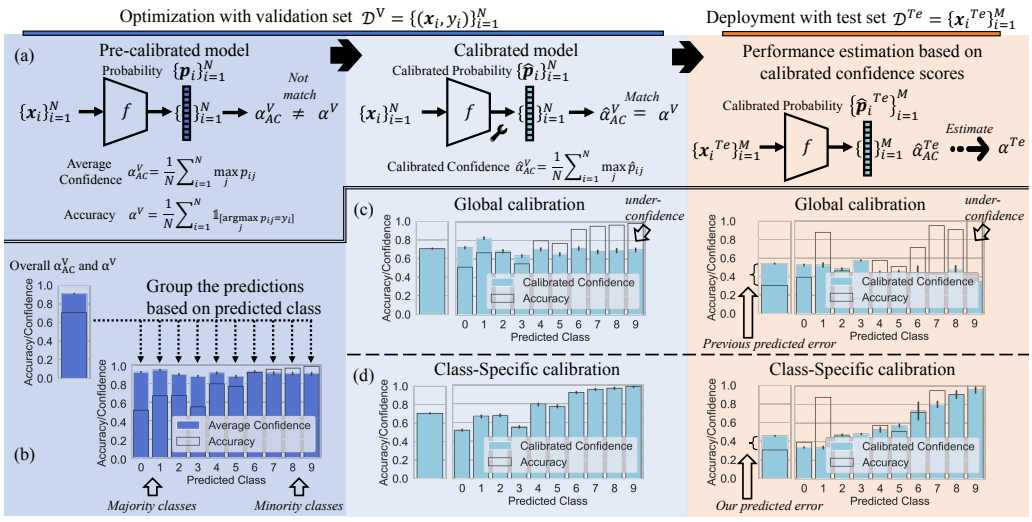


Figure 6.1: Effect of class imbalance on confidence-based model evaluation methods. (a) The pipeline for model evaluation [77, 82], which estimates the accuracy under domain shifts with average confidence (AC) after obtaining calibrated probabilities $\hat{\boldsymbol{p}}_i$. (b) The confidence scores are biased towards the majority classes. (c) Prior methods reduce all probabilities in the same manner to match overall AC with the overall accuracy, but fail to calibrate class-wise probability. (d) Our proposed method with class-specific calibration parameters successfully reduces the bias and significantly increases the model estimation accuracy. The shown example is for imbalanced CIFAR-10.

The prediction results by AC are likely to be higher than the real accuracy as modern neural networks are known to be over-confident [83]. A few methods, such as temperature scaling (TS) [83], difference of confidences (DoC) [82], and average thresholded confidence (ATC) [77], have been

proposed to calibrate the probability of a trained model with validation data with the expectation to generalize well to unseen test data, as depicted in Fig. 6.1(a). We argue that those methods do not generalize well to imbalanced datasets, which are common in medical image classification and segmentation. Specifically, previous work found that a model trained with imbalanced data is prone to predict a test sample from a minority class as a majority class, but not vice versa [147]. As a result, such models tend to have high classification accuracy but are under-confident on samples predicted as the minority classes, with opposite behaviour for the majority classes, as illustrated in Fig. 6.1(b). Current post-training calibration methods adopt the same parameter to adjust the probability, which would increase the estimation error for the minority class samples and thus, does not generalize well, as shown in Fig. 6.1(c). We introduce class-specific parameters into the performance estimation framework using adaptive optimization, alleviating the confidence bias of neural networks against minor classes, as demonstrated in Fig. 6.1(d). It should be noted that our method differs from other TS methods with class-wise parameters such as vestibular schwannoma (VS) [83], class-distribution-aware TS [109] and normalized calibration (NORCAL) [189] as 1) our parameters are optimized so that class-specific confidence matches class-specific accuracy, whereas their parameters are optimized such that overall-confidence matches overall-accuracy; 2) we determine the calibrated class based on prediction instead of ground truth, therefore the model predictions would not be affected by the calibration process; 3) we apply the proposed methods to model evaluation; 4) we also consider other calibration methods than TS.

This study sheds new light on the problem of model performance estimation under domain shifts in the absence of ground truth with the following contributions: 1) We find that existing solutions encounter difficulties in real-world datasets as they do not consider the confidence bias caused by class imbalance; 2) We propose class-specific variants for three state-of-the-art (SOTA) approaches including TS, DoC and ATC to obtain class-wise calibration; 3) We extend the confidence-based model performance framework to segmentation and prediction of Dice similarity coefficient (DSC); 4) Extensive experiments performed on four tasks with a total of 310 test settings show that the proposed methods can consistently and significantly improve the estimation accuracy compared to prior methods. Our methods can reduce the estimation error by 18% for skin lesion classification under natural domain shifts and reduce the estimation error by 50% in segmentation.

6.2 Related Works

Many methods have been proposed to tackle the problem of model estimation and they can be categorized into three groups.

The first line of research is based on the calculation of a distance metric between the training and test datasets [45, 57]. The model performance can be obtained by re-weighting the validation accuracy [45] or training a regression model [57].

The second type of methods is based on reverse testing which trains a new reverse classifier using test images and predicted labels as pseudo ground truth [71, 230]. The model performance is then determined by the prediction accuracy of the reverse classifier on the validation data.

The third group of methods is based on model confidence scores [77, 82, 83]. The accuracy of a trained model can be estimated with the AC which is the mean value of the maximum softmax confidence on the target data. Previous studies have shown that confidence-based methods show better performance than the first two methods [68, 77] and are efficient as they do not need to train additional models. We focus on confidence-based methods in this study.

6.3 Method

6.3.1 Problem Setup

We consider the problem of c-class image classification. We train a classification model f that maps the image space $\mathcal{X} \subset \mathbb{R}^d$ to the label space $\mathcal{Y} = \{1, 2, \ldots, c\}$. We assume that two sets of labeled images: $\mathcal{D}^{Tr} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{I}$ and $\mathcal{D}^{V} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ drawn from the *same* source domain are available for model training and validation. After being trained, the model is applied to a target domain $\mathcal{D}^{Te}$ whose distribution is *different* from the source domain, which is typical in the real-world applications. We aim to estimate its accuracy α^{Te} on the target domain to support decision making when there is only a set of test images $\mathcal{D}^{Te} = \{\boldsymbol{x}_i^{Te}\}_{i=1}^{M}$ available *without any ground-truth*.

6.3.2 A Unified Test Accuracy Framework Based on Model Calibration

To achieve this goal, we first provide a unified test accuracy estimation framework that is compatible with existing confidence-based methods. Let $\boldsymbol{p}_i$ be the predicted probability for input $\boldsymbol{x}_i$, which is obtained by applying the softmax function to the network output logit: $\boldsymbol{z}_i = f(\boldsymbol{x}_i)$. The predicted probability for the j'th class is computed as $p_{ij} = \sigma(\boldsymbol{z}_i)_j = \frac{\mathrm{e}^{z_{ij}}}{\sum_{j=1}^{c} \mathrm{e}^{z_{ij}}}$, and the predicted class y_i' is associated with the one of the largest probability: $y_i' = \operatorname{argmax}_j p_{ij}$. The test accuracy on the test set α^{Te} can be estimated via AC [83]: $\alpha_{AC}^{Te} = \frac{1}{N} \sum_{i=1}^{N} \max_j p_{ij}^{Te}$, which is an average of the probability of the predicted class j: $\max_j p_{ij}^{Te}$ for every test image $\boldsymbol{x}_i^{Te}$. α_{AC}^{Te} is useful to estimate the test accuracy α^{Te} as we assume that the model should be aware of the possibility of making wrong predictions.

Yet, directly computing over the softmax output p_{ij}^{Te} for estimation can be problematic as the predicted probability cannot reflect the real accuracy with modern neural networks [83]. It is essential to use calibrated probability $\hat{p}_{ij}^{Te}$ instead to estimate model accuracy with $\hat{\alpha}_{AC}^{Te} = \frac{1}{N} \sum_{i=1}^{N} \max_j \hat{p}_{ij}^{Te}$. Ideally, the calibrated probability $\hat{p}_{ij}^{Te}$ is supposed to perfectly reflect the expectation of the real accuracy. To adjust $\hat{p}_{ij}$, it is a common practice to use validation accuracy, which is defined as $\alpha^{V} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{[y_i' = y_i]}$[1], as an surrogate objective to find the solution as there is no ground truth labels for test data [68, 77, 82]. On top of this pipeline which is also illustrated in Fig. 6.1(a), we will introduce state-of-the-art calibration methods including TS, DoC, and ATC and our proposed class-specific modifications of them (cf. Fig. 6.2), alleviating the calibration bias caused by the common class imbalance issue in medical imaging mentioned earlier.

[1] $\mathbb{1}$ is an indicator function which is equal to 1 when the underlying criterion satisfies and 0 otherwise.

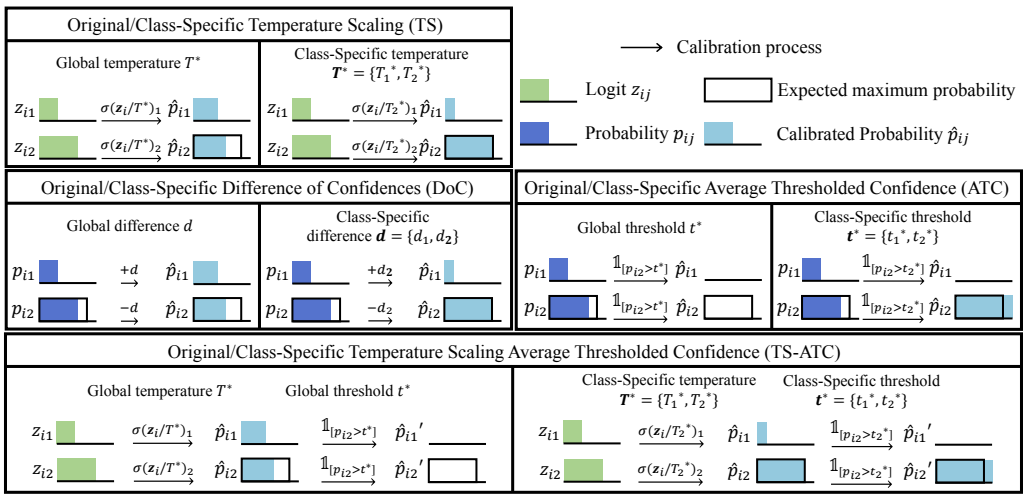


Figure 6.2: The illustration of proposed Class-Specific modifications for four existing model evaluation methods. We show the calibration process of an under-confident prediction made for sample from minority class $c = 2$. Prior calibration methods use a global parameter for all classes, which leads to sub-optimal calibration and therefore bias for the minority class. The proposed variants adapt separate parameters per class, enabling improved, class-wise calibration. Please refer to the content of text for details.

6.3.3 Class-Specific Temperature Scaling

The TS method [83] rescales the logits using temperature parameter T. Calibrated probability is obtained with $\hat{p}_{ij} = \sigma(\boldsymbol{z}_i/T)_j$. To find T such that the estimated accuracy will approximate the real accuracy, we obtain optimal T^* using $\mathcal{D}^V$ with:

$$T^* = \underset{T}{\operatorname{argmin}} \left| \frac{1}{N} \sum_{i=1}^{N} \max_j \sigma(\frac{\boldsymbol{z}_i}{T})_j - \alpha^V \right|. \tag{6.1}$$

To enable class-specific calibration, we extend temperature to vector $\boldsymbol{T}$ with c elements, where j'th element T_j corresponds to the j'th class. The calibrated probability is then written as $\hat{p}_{ij} = \sigma(\boldsymbol{z}_i/T^*_{y'_i})_j$, where T^*_j is optimized with:

$$T^*_j = \underset{T_j}{\operatorname{argmin}} \left| \frac{1}{N_j} \sum_{i=1}^{N} \mathbb{1}_{ij} \max_j \sigma(\frac{\boldsymbol{z}_i}{T_j})_j - \alpha^V_j \right|, \tag{6.2}$$

where $\mathbb{1}_{ij} = \mathbb{1}_{[y'_i = j]}$ is 1 if the predicted most probable class is j and 0 otherwise, $N_j = \sum_{i=1}^{N} \mathbb{1}_{ij}$ and α^V_j is the accuracy for class j on the validation set which is calculated as $\alpha^V_j = \frac{1}{N_j} \sum_{i=1}^{N} \mathbb{1}_{ij} \mathbb{1}_{[y'_i = y_i]}$. This way we ensure optimized $\boldsymbol{T}^*$ leads to $\hat{\boldsymbol{p}}_i$ matching the class-wise accuracy.

6.3.4 Class-Specific Difference of Confidences

The DoC method [82] adjusts p_i by subtracting the difference $d = \alpha^V_{AC} - \alpha^V$ between the real accuracy α^V and α^V_{AC} from the maximum probability:

$$\hat{p}_{ij} = \begin{cases} p_{ij} - d & \text{if} \quad y'_i = j, \\ p_{ij} + \frac{d}{c-1} & \text{otherwise.} \end{cases} \tag{6.3}$$

The two parts of Eq. 6.3 ensure that the class-wise re-calibrated probabilities remain normalized. We propose a class-specific DoC method, where we define the difference vector $\boldsymbol{d}$ with c elements, with its j'th element d_j corresponding to the j'th class's difference. The class-wise calibrated probability $\hat{p}_{ij}$ is then obtained via:

$$\hat{p}_{ij} = \begin{cases} p_{ij} - d_j & \text{if} \quad y'_i = j, \\ p_{ij} + \frac{d_{y'}}{c-1} & \text{otherwise,} \end{cases} \tag{6.4}$$

where $d_j = \frac{1}{N_j} \sum_{i=1}^{N} \mathbb{1}_{ij} \max_j p_{ij} - \alpha^V_j$ is the difference between the average predicted probability for samples predicted that they belong to class j and α^V_j. In this way, the probabilities of samples which are predicted as the minority classes would not be affected by the majority classes and thus be less biased.

6.3.5 Class-Specific Average Thresholded Confidence

The original ATC [77] proposes predicting the accuracy as the portion of samples with $\boldsymbol{p}_i$ that is higher than a learned threshold t. t is obtained via:

$$t^* = \underset{t}{\operatorname{argmin}} \left| \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{[\max_j p_{ij} > t]} - \alpha^V \right|. \tag{6.5}$$

We can then use t^* to compute $\hat{p}_{ij} = \mathbb{1}_{[p_{ij} > t^*]}$ for each test sample to calculate the estimated test accuracy, $\hat{\alpha}^{Te}_{AC}$, defined previously. We should note that for ATC, $\hat{p}_{ij}$ is no longer a calibrated probability but a discrete number $\in \{0, 1\}$. For class-specific calibration, we propose learning a set of thresholds given by vector $\boldsymbol{t}$ with c elements, one per class, where j'th element is obtained by:

$$t^*_j = \underset{t_j}{\operatorname{argmin}} \left| \frac{1}{N_j} \sum_{i=1}^{N} \mathbb{1}_{ij} \mathbb{1}_{[max_k p_{ik} > t_j]} - \alpha^V_j \right|. \tag{6.6}$$

We can then calibrate probabilities as $\hat{p}_{ij} = \mathbb{1}_{[p_{ij} > t^*_{y'_i}]}$ based on predicted class y'_i. Hence the probabilities of samples that are predicted as different classes are discretized using different thresholds

$t^*_{y'_i}$ to compensate the bias incurred by class imbalance. ATC can be combined with TS, by applying it to $\hat{\boldsymbol{p}}_i$ which is calibrated by TS, and we denote this method as TS-ATC. By combining our class-specific modifications of TS and ATC, we similarly obtain the class-specific variant CS TS-ATC (cf. Fig. 6.2).

6.3.6 Extension to Segmentation

We propose extending confidence-based model evaluation methods to segmentation by predicting the DSC that the model would achieve on target data, instead of the accuracy metric that is more appropriate for classification. For this purpose, we slightly modify notation and write that the validation dataset consists of Z segmentation cases as $\mathcal{D}^V = \{\mathcal{S}_z\}_{z=1}^Z$, with $\mathcal{S}_z$ all the training pairs for the case z consisting of totally n pixels, defined as $\mathcal{S}_z = \{(\boldsymbol{x}_{zi}, y_{zi})\}_{i=1}^n$, where $\boldsymbol{x}_{zi}$ is an image patch sampled from case z and y_{zi} is the ground truth class of its central pixel. We then denote p_{zij} as the predicted probability of sample $\boldsymbol{x}_{zi}$ for the j'th class and its calibrated version as $\hat{p}_{zij}$. The estimated DSC can then be calculated using the predicted probabilities, in a fashion similar to the common soft DSC loss [171, 175]. Specifically, DSC_j, which is real DSC of $\mathcal{D}^V$ for the j'th class, is estimated with:

$$\text{sDSC}_j = \frac{1}{Z}\sum_{z=1}^{Z}\frac{2\sum_{i=1}^{n}\mathbb{1}_{[\text{argmax}_k\, p_{zik}=j]}\hat{p}_{zij}}{\sum_{i=1}^{n}\mathbb{1}_{[\text{argmax}_k\, p_{zik}=j]} + \sum_{i=1}^{n}\hat{p}_{zij}}. \tag{6.7}$$

For the CS TS, CS ATC and CS TS-ATC, we first calibrate p_{zij} of the background class by matching class-wise average confidence and accuracy α_j^V, then calibrate p_{zij} of the foreground class j by matching sDSC_j and DSC_j, in order to adjust the model to fit the metric of DSC. We calibrate the background class according to accuracy instead of DSC because the sDSC_j is always very high and cannot be easily decrease to match DSC_j, which we found makes optimization to match them meaningless. For CS DoC, as d_j cannot be calculated with the difference between sDSC_j and DSC_j, we keep $\hat{p}_{zij} = p_{zij}$ and estimate DSC on the target dataset $\mathcal{D}^{Te}$ on j'th class simply by $\text{DSC}^{Te}_{CSDoCj} = \text{DSC}_j - \text{sDSC}_j + \text{sDSC}^{Te}_j$.

6.4 Experiments

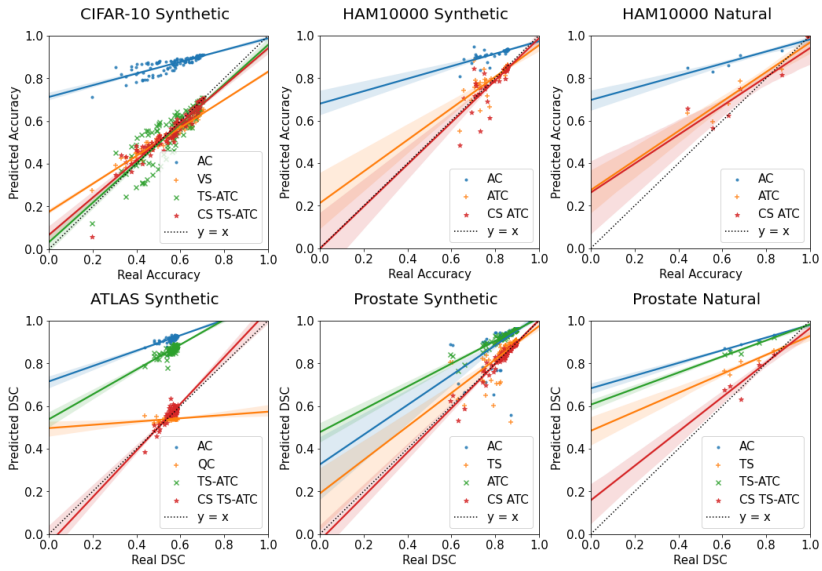


Figure 6.3: Real versus predicted accuracy/DSC for classification/segmentation tasks respectively. Each point represents model performance on the unseen target data. We plot a baseline method (AC) in blue, the best among previous methods in orange, the best among the proposed methods in red and its class-agnostic counterpart in green. Best methods determined based on mean absolute error (MAE).

6.4.1 Setting

In our experiments, we first train a classifier f with $\mathcal{D}^{Tr}$, then we make predictions on $\mathcal{D}^{V}$ and optimize the calibration parameters based on model output and the validation labels. Finally, we evaluate our model on $\mathcal{D}^{Te}$ and estimate the quality of the predicted results based on $\hat{\boldsymbol{p}}_i$.

In addition to the confidence-based methods described in Sec. 6.3, for the segmentation tasks, we also compare with a neural network based quality control (QC) method [203]. Following [203], we train a three-dimensional (3D) ResNet-50 that takes as input the image and the predicted segmentation mask and predicts the DSC of the predicted versus the manual segmentation. For fair comparison, we train the QC model with the same validation data we use for model calibration.

We implement VS [83] as a base line, which is similar to TS but adopt different temperature parameter T_j and bias b_j for different classes. Specifically, the calibrated probability is written as $\hat{p}_{ij} = \sigma(\boldsymbol{z}_i/T_j + b_j)_j$. Although VS utilizes class-specific parameters, it is optimized to match the global accuracy and the global confidence, similar to Eq. 6.1.

We also implement NORCAL [189], of which temperature parameters are scaled based on the class

frequency. Similar idea is adopted in class-distribution aware TS [109]. Specifically, the class-specific temperature parameter of class j is $T_j = TN/N_j$, where N is the number of the data samples in the training dataset and N_j is the number of samples belonging to class j.

6.4.2 Datasets

We visualize all the datasets containing domain shifts in Fig. 6.4.

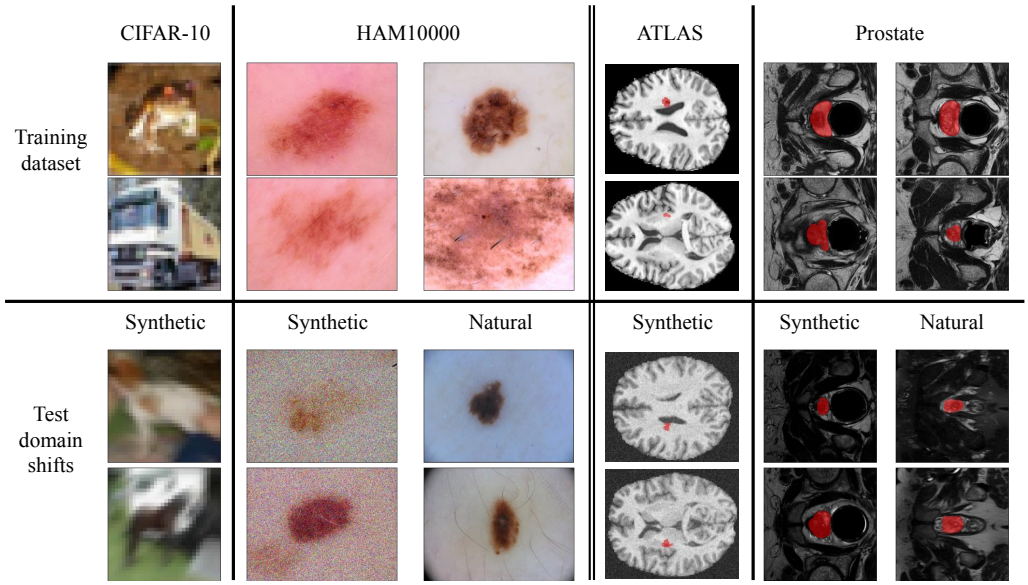


Figure 6.4: Visualization of the datasets containing domain shifts.

6.4.2.1 Imbalanced CIFAR-10

We train a ResNet-32 [93] with imbalanced CIFAR-10 [126], using imbalance ratio of 100 following [30]. Specifically, we intentionally resample the training datatset to have {5000, 2997, 1796, 1077, 645, 387, 232, 139, 83, 50} samples for each class. We employ synthetic domain shifts using CIFAR-10-C [97] that consists of 95 distinct corruptions.

6.4.2.2 Skin lesion classification

We train ResNet-50 for skin lesion classification with $c = 7$ including melanocytic nevi (nv), melanoma (mel), basal cell carcinoma (bcc), dermatofibroma (df), nenign keratosis (bkl), vascular (vasc) and actinic keratoses (akiec), following [168, 207]. We randomly select 70% data from HAM10000 [228] as the training source domain and leave the rest as validation. We also create synthetic domain shifts based on the validation data following [97], obtaining 38 synthetic test datasets with domain shifts. We additionally collect 6 publicly available skin lesion datasets which were collected

in different institutions including BCN [50], VIE [205], MSK [86], UDA [86], D7P [121] and PH2 [172] following [250]. The detailed dataset statistics are summarized in Table 6.1.

Table 6.1: Dataset statistics of skin lesion classification datasets.

	nv (%)	mel (%)	bcc (%)	df (%)	bkl (%)	vasc (%)	akiec (%)	total
HAM [228]	6705 (67)	1113 (11)	514 (5)	115 (1)	1099 (11)	142 (1)	327 (3)	10015
BCN [50]	4206 (34)	2857 (23)	2809 (23)	124 (1)	1138 (9)	111 (1)	1168 (9)	12413
VIE [205]	4331 (99)	34 (1)	0	0	0	0	0	4365
MSK [86]	2202 (62)	826 (23)	30 (1)	5 (<1)	470 (13)	0	7 (<1)	3540
UDA [86]	408 (67)	193 (31)	3 (<1)	2 (<1)	7 (1)	0	0	613
D7P [121]	1150 (60)	501 (26)	84 (4)	40 (2)	90 (5)	58 (3)	0	1923
PH2 [172]	160 (80)	40 (20)	0	0	0	0	0	200

6.4.2.3 Brain lesion segmentation

We apply nnU-Net [106] for all the segmentation tasks. We evaluate the proposed methods with brain lesion segmentation based on T1-weighted magnetic resonance (MR) images using data from anatomical tracings of lesions after stroke (ATLAS) [152] for which classes are highly imbalanced. We randomly select 145 cases for training and leave 75 cases for validation. We create synthetic domains shifts by applying 83 distinct transformations, including spatial, appearance and noise operations to the validation data. Details on implementation are in Fig. 5.2.

6.4.2.4 Prostate segmentation

We also evaluate on the task of cross-site prostate segmentation based on T2-weighted MR images following [158]. We use 30 cases collected with 1.5T magnetic resonance imaging (MRI) machines from [21] as the source domain. We randomly select 20 cases for training and 10 cases for validation. We create synthetic domains shifts using the same 83 transformations as for brain lesions. As target domain, we use the 5 datasets described in [21, 137, 156], collected from different institutions using scanners of different field strength and manufacturers.

6.4.3 Results and Discussion

We calculate the MAE and R2 Score between the predicted and the real performance (accuracy or DSC). We summarize MAE results in Table 6.2 and R2 Score results in Table 6.3. We also summarize the boxplots of the results in Fig. 6.5. We show scatter plots of predicted versus real performance in Fig. 6.3. Each reported result is the average of two experiments with different random seeds.

We observe in Table 6.2 that our proposed modifications for class-specific calibration improve all baselines. Improvements are especially profound in segmentation tasks, where class imbalance is very high. It might be because previous model calibration methods bias towards the background class significantly as a result of class imbalance. These results clearly demonstrate the importance of accounting for class imbalance in the design of performance-prediction methods, to counter biases induced by the data. Fig. 6.3 shows that the predicted performance by AC baseline is always higher than real performance because neural networks are commonly over-confident [83]. QC is effective in predicting real DSC under synthetic domain shifts but under-performs with natural domain shifts. Results in both Table 6.2 and Fig. 6.3 show that CS ATC (or CS TS-ATC) consistently performs best, outperforming prior methods by a large margin.

Table 6.2: Evaluation on different tasks under varied types of domain shifts based on MAE. Lower MAE is better. Best results with lowest MAE in **bold**. Class-specific calibration as proposed (CS methods) improves all baselines. This is most profound in segmentation tasks, which present extreme class imbalance.

Task	Classification			Segmentation		
Training dataset	CIFAR-10	HAM10000		ATLAS	Prostate	
Test domain shifts	Synthetic	Synthetic	Natural	Synthetic	Synthetic	Natural
AC	31.3 ± 8.2	12.3 ± 5.1	20.1 ± 13.4	35.6 ± 2.1	8.7 ± 4.9	18.7 ± 5.9
QC [203]	−−	−−	−−	3.0 ± 1.7	5.2 ± 6.6	19.3 ± 7.1
TS [83]	5.7 ± 5.6	3.9 ± 4.3	12.1 ± 8.3	9.7 ± 2.5	3.7 ± 5.4	9.2 ± 4.9
VS [83]	3.8 ± 2.1	4.2 ± 4.2	13.6 ± 9.6	11.4 ± 2.5	4.8 ± 5.1	11.2 ± 4.9
NORCAL [189]	7.6 ± 3.8	4.2 ± 4.6	13.7 ± 9.6	6.7 ± 2.4	5.8 ± 5.7	7.3 ± 4.7
CS TS	$5.5^{\sim} \pm 5.6$	$3.7^{\sim} \pm 4.0$	$11.9^{\sim} \pm 8.0$	$1.6^{**} \pm 1.8$	$3.0^{**} \pm 5.7$	$7.8^{**} \pm 4.8$
DoC [82]	10.8 ± 8.2	4.6 ± 5.0	15.3 ± 9.7	4.2 ± 3.2	3.7 ± 5.8	13.9 ± 6.5
CS DoC	$9.4^{**} \pm 7.2$	$4.5^{\sim} \pm 4.9$	$14.7^{*} \pm 9.2$	$\mathbf{1.3}^{**} \pm \mathbf{1.9}$	$3.5^{*} \pm 6.1$	$12.1^{*} \pm 5.9$
ATC [77]	4.6 ± 4.4	3.4 ± 3.9	7.1 ± 6.3	30.4 ± 1.8	8.6 ± 3.3	16.7 ± 5.3
CS ATC	$2.8^{**} \pm 2.9$	$\mathbf{3.3}^{\sim} \pm \mathbf{4.8}$	$\mathbf{5.8}^{\sim} \pm \mathbf{7.6}$	$1.6^{**} \pm 1.5$	$\mathbf{1.1}^{**} \pm \mathbf{1.7}$	$4.3^{**} \pm 2.2$
TS-ATC [77, 83]	5.3 ± 3.9	4.2 ± 4.2	7.3 ± 7.1	30.4 ± 1.8	8.5 ± 3.3	16.7 ± 5.3
CS TS-ATC	$\mathbf{2.7}^{**} \pm \mathbf{2.3}$	$4.2^{\sim} \pm 5.4$	$5.9^{\sim} \pm 8.4$	$\mathbf{1.3}^{**} \pm \mathbf{1.4}$	$1.2^{**} \pm 1.7$	$\mathbf{4.2}^{**} \pm \mathbf{2.2}$

$^{*}p$-value < 0.05; $^{**}p$-value < 0.01; $^{\sim}p$-value ≥ 0.05 (compared with their class-agnostic counterparts)

Table 6.3: Evaluation on different tasks under varied types of domain shifts based on R2 Score. Higher R2 Score is better. The best results are in **bold**.

Task	Classification			Segmentation		
Training dataset	CIFAR-10	HAM10000		ATLAS	Prostate	
Test domain shifts	Synthetic	Synthetic	Natural	Synthetic	Synthetic	Natural
AC	-7.88	-3.15	-0.70	-157.94	-1.29	-4.52
QC [203]	––	––	––	-0.50	-0.63	-5.02
TS [83]	0.46	0.22	0.38	-11.59	0.02	-0.56
VS [83]	0.84	0.18	0.19	-16.14	-0.12	-1.16
NORCAL [189]	0.39	0.11	0.19	-5.26	-0.51	-0.07
CS TS	0.48	0.30	0.40	0.28	0.04	-0.19
DoC [82]	-0.55	-0.07	0.04	-2.47	-0.08	-2.38
CS DoC	-0.19	-0.04	0.13	0.30	-0.13	-1.59
ATC [77]	0.66	**0.37**	**0.74**	-114.97	-0.93	-3.39
CS ATC	0.86	0.20	0.73	0.40	**0.91**	0.67
TS-ATC [77, 83]	0.63	0.17	0.70	-114.80	-0.92	-3.37
CS TS-ATC	**0.89**	-0.10	0.70	**0.54**	0.90	**0.68**

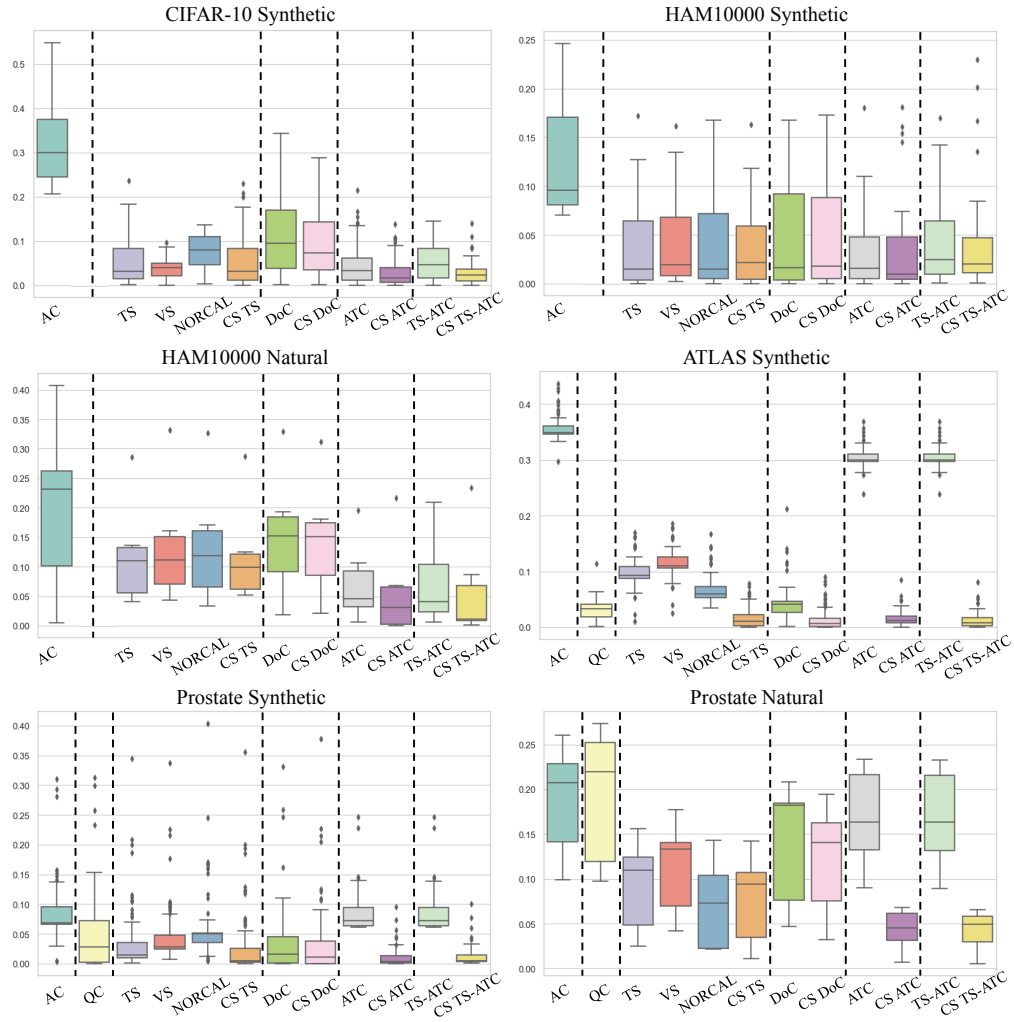


Figure 6.5: Boxplots of the MAE results for different methods. The methods with the same original formulation are grouped together.

6.5 Conclusion

In this chapter, we propose estimating model performance under domain shift for medical image classification and segmentation based on class-specific confidence scores. We derive the class-specific variants for SOTA methods to alleviate calibration bias incurred by class imbalance, which is profound in real-world medical imaging databases, and show very promising results.

We observe that our proposed method cannot improve much for the synthetic domain shifts of skin lesion classification. This is may because the synthetic transformations were initially designed for natural image classification but might not be realistic for dermatoscopic images. In the future, a fine analysis of the estimation results with class accuracy or Expected Calibration Error (ECE) could help us better understand our algorithms.

We extend the confidence scores based model estimation methods to segmentation tasks by estimating DSC. We only utilize the pixel-wise probability to estimate the segmentation accuracy. In the future, we should also consider to take account the shape information of regions of interest (ROIs) for segmentation tasks.

We study confidence scores based model estimation methods and only utilize the maximum of prediction probability after softmax as confidence scores. Future works could consider to investigate the impact of class imbalance with other kinds of confidence scores [16] such as Monte-Carlo dropout [76], TrustScore [112], DOCTOR [80] and DUQ [232].

We expect the proposed methods to be useful for safe deployment of machine learning in real world settings, for example by facilitating decisions such as whether to deploy a given, pre-trained model based on whether estimated model performance on the target data of a deployment environment meets acceptance criteria.

Chapter 7

Conclusions

This thesis is essentially concerned with the class imbalance problem of neural network training for segmentation. For this purpose, we analyze model behaviours based on empirical observations under the setting of class imbalance. Beyond the analysises of foreground overfitting and background underfitting, we propose several learning strategies to make neural network generalize better, showing promising improvements. Furthermore, we focus on accessing model performance under class imbalance and domain shifts, which is crucial for safe-critical applications such as medical image segmentation.

This chapter summarizes the main arguments and contributions we make in Chapter 3, Chapter 4, Chapter 5 and Chapter 6. Most of our findings are general for neural networks trained with data that presents class imbalance. Subsequently, we discuss some open challenges in the context of class imbalance for wider machine learning applications.

We provide the code for all of the experiments to facilitate future works:

1. Chapter 3: https://github.com/ZerojumpLine/OverfittingUnderClassImbalance
2. Chapter 4: https://github.com/ZerojumpLine/CoLab
3. Chapter 5: https://github.com/ZerojumpLine/JCSAugment
4. Chapter 6: https://github.com/ZerojumpLine/ModelEvaluationUnderClassImbalance

7.1 Summary of Contributions

Class imbalance widely exists in the datasets for semantic segmentation and restricts the performance of neural networks, especially in medical imaging. Analyzing the effects of class imbalance on model training can help us identify and tackle the limitations of neural networks trained with imbalanced training data. In this thesis, we systematically look into the effects of class imbalance on model behaviors in Chapter 3 and Chapter 4 with empirical experiments. Specifically, we trained neural networks with different amounts of data and different kinds of context labels. We monitored the logit distribution when the segmentation models processing training and unseen test samples. In addition, we extended our findings to domain shifts in Chapter 6. In summary, we shed new lights on the analysis of class imbalance by making following contributions:

- With the examination of four segmentation tasks and two popular segmentation models, we find that when machine learning model were trained with limited data with class imbalance, it consistently shows decreased segmentation performance with reduced sensitivity. Furthermore, with the comparison of the logit distributions when the model processes training data or unseen test data, we conclude that neural networks tend to map the foreground samples close to the decision boundaries as the result of overfitting under class imbalance.

- Via the inspection of six medical image segmentation tasks, we find that neural networks systematically over-segment small regions of interest (ROIs) with diverse background. Moreover, we find that neural networks cannot map the heterogeneous background samples to compact clusters in feature space. As a result, the logit activations of the background samples approach and even move across the decision boundary. Empirically, we observe that model can learn stable representation by manually decomposing the background class into several subclasses.

- Based on the observations of model performance under domain shifts when model trained with data that presents class imbalance, we find that neural networks are prone to predicting a sample from a minority class as a majority class under domain shifts. We find this is consistent with our observations of overfitting under class imbalance, which may become more substantial under domain shifts. As a result, models are likely to be under-confident about the minority classes and over-confident about the majority classes.

Based on our findings and analysis, we propose new learning strategies for neural networks to improve segmentation performance. We advance neural networks with novel loss functions, regularization techniques and multi-task learning, showing promising improvements for varied medical image segmentation tasks. Specifically, we tackle overfitting of foreground samples and underfitting of background samples in Chapter 3 and Chapter 4. We propose to alleviate both overfitting and underfitting under class imbalance with by learning class-specific data augmentation strategies in Chapter 5. We summarize our contributions as follow:

- We introduce new asymmetric variants of popular loss functions and regularization techniques including a large margin loss, focal loss, adversarial training, mixup and data augmentation, which are explicitly designed to counter logit shift of the minority classes. We validate the proposed methods with two convolutional neural network (CNN) models and four challenging segmentation tasks. We find that our methods can explicitly reduce the logit shifts of the under-represented classes at test time, yielding consistently improved segmentation accuracy for small ROI segmentation with higher sensitivity when compared to baselines.

- We propose context label learning (CoLab) to improve the context representations by decomposing the background class into several subclasses. Specifically, we train an auxiliary network as a task generator, along with the primary segmentation model, to automatically generate context labels that positively affect the ROI segmentation accuracy. We validate CoLab with six segmentation datasets which contain heterogeneous background samples, showing improved segmentation accuracy with higher precision. CoLab has similar and sometimes better effect in improving segmentation accuracy when compared with human-defined context labels.

- We propose to adopt different kinds of training-time data augmentation (TRA) for training samples from different classes, effectively extending the training data distribution and alleviating the class imbalance issue. In addition, we jointly optimize TRA and test-time data augmentation (TEA), which are closely connected as both aim to align the training and test data distribution but were so far considered separately in previous works. We validate the proposed data augmentation framework with extensive experiments based on four segmentation datasets and two CNN models, consistently improving the segmentation performance when compared to existing solutions.

The success of neural networks relies on the assumption that the model is deployed and trained with data from the same distribution. However, the assumption does not always hold for real-world applications where neural networks could perform badly in an unexpected way. Estimating the performance under domain shifts is important for building reliable machine learning models. This thesis investigates the problem of performance estimation with the existence of domain shifts in Chapter 6. Specifically, we propose to reduce the class bias in confidence based model estimation methods with class-specific calibration parameters. In sum, we make following contributions:

- We identify the limitations of current confidence-based performance estimation methods for real-world datasets that present class imbalance. We find existing post-training calibration methods do not account for the bias induced by class imbalance. Specifically, current methods adopt the same parameters for different classes without considering that domain shifts are prone to making neural networks under-confident about the predictions of the minority classes. Consequently, such approaches would increase the estimation error for the minority classes and cannot generalize well.

- We derive class-specific modifications of confidence-based model evaluation methods including temperature scaling (TS), difference of confidences (DoC) and average thresholded confidence (ATC) which adapt separate parameters per class, enabling class-wise calibration. We validate our methods with four tasks across image classification and image segmentation. We find that the proposed modifications can consistently improve the estimation accuracy when models are trained with imbalanced data.

7.2 Open Challenges and Future Outlook

In this work, we find that class imbalance would have a negative impact on the learning of neural networks for image segmentation. We think that the class imbalance problem would also affect other machine learning tasks as real-world data would always follows a long-tailed distribution [65, 164]. Here, we discuss some research directions to advance neural networks in other computer vision and healthcare problems, where class imbalance could also affect the performance.

7.2.1 Class Imbalance in Other 3D Vision Tasks

Compared with two-dimensional (2D) problems, the class imbalance ratio becomes even more extreme in three-dimensional (3D) tasks as small objects account for a small proportion in the 3D space. In this case, small 3D objects which are rare or indiscernible are easier to be under-represented by machine learning models. For example, learning-based multi-view stereo methods do not perform well on the depth estimation of textureless and reflective surfaces [7] which are hard for multi-view matching. Eye localization and alignment in 3D fetal brain ultrasound [181] is difficult as eyes only appear in a small proportion of the sampled slices.

A nature extension of our study is to analyze the failure reasons caused by class imbalance beyond classification problems in 3D vision such as 3D reconstruction, detection and alignment, from the perspective of model behavior. A systematic comparison of the network representations of under-represented classes and those of well-represented classes could be considered to complement our findings. Specifically, the under-represented regions could be determined with uncertainty. For the task of 3D reconstruction, the knowledge from other samples could be utilized to provide customized inductive biases [249] for better representations. The investigation of an active model to guide the practitioner to collect additional 3D data for better under-represented classes performance is also a promising research direction.

Video includes 3D scenes as time can be taken as the third dimension. The class imbalance problem exists in the problem of action recognition as the number of video clips per class would be different, and could be a bottleneck as rare classes are found to have a poor yield rate [31]. Similarly, some anatomical categories are imbalanced in fetal ultrasound scan videos [213], affecting the classification performance of minority classes. The investigation of the different classes under class imbalance for video recognition and the development of algorithms to improve the yield rate of rare classes can be considered as a future direction of our work. Specifically, the comparison of the local and generic representations learned for different classes could be made by performing gradient descent on random inputs [72]. Our analysis of class imbalance could be also extended to the task of video retrieval [162], which also demands robust representations of the minority classes. As videos contain motion information and the networks can learn local representations that are highly class-specific [72], it could also be interesting to study how the motion information

connect to class imbalance and how it can affect the network behavior. The representation of rare classes could be enhanced by augmenting with synthesized motion, by means of mixup [261] or causal intervention [260].

7.2.2 Class Imbalance in Self-Supervised Learning

Recently, self-supervised learning (SSL) has been found to be an effective pre-training approach to achieve better network weight initialization. Current SSL methods, such as contrastive learning [92], would adopt the same training strategy for all the samples, despite their categories. As previous studies found that incorporating semantic class into SSL [87] can largely boost the performance of downstream tasks, there could be many valuable and unexplored questions under the setting of class imbalance: Can we improve SSL for class imbalanced training dataset by cluster-specific weighting or implicitly encouraging the number of cluster samples based on class imbalance ratio? How does the representation obtained with SSL differ between the majority classes and the minority classes?

Analysis could be started with visual representation learning of single modality and then extended to the settings of multi-modality contrast learning such as image and motion in video representation [87] as well as video and speech in ultrasound imaging [113], which are more challenging as the implicit class distribution may differ across modalities. In a broader context, further research is needed to facilitate the pre-training using SSL on large scale imbalanced datasets containing paired or unpaired multi-modality images, which is important for medical imaging as the labels are scarce and medical images are always collected in the form of different modalities.

7.2.3 Class Imbalance in Multi-Modality Learning

Beyond image classification, machine learning models can help us build connections between images and data from other modalities, such as image and language [198] or image and patient phenotypes [8]. The class imbalance problem also exists in the problem of multi-modality learning as language and patient phenotypes could also follow long-tailed distributions. It would be potentially valuable to find a way to enhance the representation of the minority classes by complementing the information from different modalities. However, few studies focus on how to effectively fuse data from different modalities to alleviate class imbalance in multi-modality learning. A follow-up study of our class imbalance observations could be the investigation of how data from different modalities or different kinds of combinations affect model behavior. It would be particularly interesting to look into the phenome-wide associations of different diseases to discover underlying valuable imaging biomarkers. Perhaps the class imbalance in regression [201] should also be considered in this setting as data from other modalities could be continuous, such as age. For example, when the models are trained with datasets which contain more elder people but are expected to be deployed to datasets containing more younger people.

7.3 Conclusions

In this thesis, we shed new light on neural networks for semantic segmentation with systematical analysis and novel learning strategies. We propose to tackle class imbalance and domain shifts, which exist simultaneously in real world datasets and limit the performance of modern machine learning models when deployed to medical image segmentation.

Motivated by the empirically observations of model performance and model behaviours, we conclude that class imbalance causes under-segmentation because of overfitting foreground samples, while over-segmentation because of underfitting background samples. Although these biases caused by class imbalance are partially noticed by previous works, few studies identify and analyze those challenges in a systematical way.

Beyond our analysis, we propose several learning strategies to alleviate class imbalance under different experiment settings. Specifically, we have following main findings: 1) Asymmetric loss functions and regularization techniques help counter overfitting under class imbalance. 2) Context labels help alleviate underfitting under class imbalance. 3) Class-specific parameters are beneficial for improving data augmentation and tackling domain shifts.

We expect the proposed methods to be useful for robust and safe deployment of machine learning in real-world settings.

References

[1] Abraham, N. and Khan, N. M. "A novel focal tversky loss function with improved attention u-net for lesion segmentation". In: *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. IEEE. 2019, pp. 683–687.

[2] Aljabar, P., Heckemann, R. A., Hammers, A., et al. "Multi-atlas based segmentation of brain images: atlas selection and its effect on accuracy". In: *Neuroimage* 46.3 (2009), pp. 726–738.

[3] Antoniou, A., Storkey, A., and Edwards, H. "Data augmentation generative adversarial networks". In: *arXiv preprint arXiv:1711.04340* (2017).

[4] Arthur, D. and Vassilvitskii, S. *k-means++: The advantages of careful seeding*. Tech. rep. Stanford, 2006.

[5] Attiyeh, M. A., Chakraborty, J., Doussot, A., et al. "Survival prediction in pancreatic ductal adenocarcinoma by quantitative computed tomography image analysis". In: *Annals of surgical oncology* 25.4 (2018), pp. 1034–1042.

[6] Attiyeh, M. A., Chakraborty, J., Gazit, L., et al. "Preoperative risk prediction for intraductal papillary mucinous neoplasms by quantitative CT image analysis". In: *Hpb* 21.2 (2019), pp. 212–218.

[7] Bae, G., Budvytis, I., and Cipolla, R. "Multi-View Depth Estimation by Fusing Single-View Depth Probability with Multi-View Geometry". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 2842–2851.

[8] Bai, W., Suzuki, H., Huang, J., et al. "A population-based phenome-wide association study of cardiac and aortic structure and function". In: *Nature medicine* 26.10 (2020), pp. 1654–1662.

[9] Bai, Y., Chen, M., Zhou, P., et al. "How Important is the Train-Validation Split in Meta-Learning?" In: *International Conference on Machine Learning*. PMLR. 2021, pp. 543–553.

[10] Bakas, S., Akbari, H., Sotiras, A., et al. "Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features". In: *Sci. Data* 4 (2017), p. 170117.

[11] Bartlett, P. L. and Mendelson, S. "Rademacher and Gaussian complexities: Risk bounds and structural results". In: *Journal of Machine Learning Research* 3.Nov (2002), pp. 463–482.

[12] Baxter, J. "Learning internal representations". In: *Proceedings of the eighth annual conference on Computational learning theory*. ACM. 1995, pp. 311–320.

[13] Bello, I., Zoph, B., Vasudevan, V., et al. "Neural optimizer search with reinforcement learning". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 459–468.

[14] Ben-David, S., Blitzer, J., Crammer, K., et al. "A theory of learning from different domains". In: *Machine learning* 79.1 (2010), pp. 151–175.

[15] Bengio, Y. "Gradient-based optimization of hyperparameters". In: *Neural computation* 12.8 (2000), pp. 1889–1900.

[16] Bernhardt, M., Ribeiro, F. D. S., and Glocker, B. "Failure Detection in Medical Image Classification: A Reality Check and Benchmarking Testbed". In: *arXiv preprint arXiv:2205.14094* (2022).

[17] Berthelot, D., Carlini, N., Goodfellow, I., et al. "Mixmatch: A holistic approach to semi-supervised learning". In: *arXiv preprint arXiv:1905.02249* (2019).

[18] Bilic, P., Christ, P. F., Vorontsov, E., et al. "The liver tumor segmentation benchmark (lits)". In: *arXiv preprint arXiv:1901.04056* (2019).

[19] Billot, B., Greve, D. N., Puonti, O., et al. "Synthseg: Domain randomisation for segmentation of brain mri scans of any contrast and resolution". In: *arXiv preprint arXiv:2107.09559* (2021).

[20] Bishop, C. M. et al. *Neural networks for pattern recognition*. Oxford university press, 1995.

[21] Bloch, N., Madabhushi, A., Huisman, H., et al. "NCI-ISBI 2013 challenge: automated segmentation of prostate structures". In: *The Cancer Imaging Archive* 370 (2015).

[22] Boykov, Y., Veksler, O., and Zabih, R. "Fast approximate energy minimization via graph cuts". In: *IEEE Transactions on pattern analysis and machine intelligence* 23.11 (2001), pp. 1222–1239.

[23] Bragman, F. J., Tanno, R., Ourselin, S., et al. "Stochastic filter groups for multi-task cnns: Learning specialist and generalist convolution kernels". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 1385–1394.

[24] Breiman, L. "Bagging predictors". In: *Machine learning* 24.2 (1996), pp. 123–140.

[25] Breiman, L. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.

[26] Brosch, T., Tang, L. Y., Yoo, Y., et al. "Deep 3D convolutional encoder networks with shortcuts for multiscale feature integration applied to multiple sclerosis lesion segmentation". In: *IEEE transactions on medical imaging* 35.5 (2016), pp. 1229–1239.

[27] Buda, M., Maki, A., and Mazurowski, M. A. "A systematic study of the class imbalance problem in convolutional neural networks". In: *Neural Networks* 106 (2018), pp. 249–259.

[28] Cai, H., Chen, T., Zhang, W., et al. "Efficient architecture search by network transformation". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[29] Cai, J., Wang, Y., and Hwang, J.-N. "ACE: Ally Complementary Experts for Solving Long-Tailed Recognition in One-Shot". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 112–121.

[30] Cao, K., Wei, C., Gaidon, A., et al. "Learning imbalanced datasets with label-distribution-aware margin loss". In: *Advances in Neural Information Processing Systems*. 2019, pp. 1567–1578.

[31] Carreira, J., Noland, E., Hillier, C., et al. "A short note on the kinetics-700 human action dataset". In: *arXiv preprint arXiv:1907.06987* (2019).

[32] Caruana, R. "Multitask learning". In: *Machine learning* 28.1 (1997), pp. 41–75.

[33] Castro, D. C., Walker, I., and Glocker, B. "Causality matters in medical imaging". In: *Nature Communications* 11.1 (2020), pp. 1–10.

[34] Chaitanya, K., Karani, N., Baumgartner, C. F., et al. "Semi-supervised task-driven data augmentation for medical image segmentation". In: *Medical Image Analysis* 68 (2021), p. 101934.

[35] Chakraborty, J., Midya, A., Gazit, L., et al. "CT radiomics to predict high-risk intraductal papillary mucinous neoplasms of the pancreas". In: *Medical physics* 45.11 (2018), pp. 5019–5029.

[36] Chapelle, O., Scholkopf, B., and Zien, A. "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]". In: *IEEE Transactions on Neural Networks* 20.3 (2009), pp. 542–542.

[37] Chawla, N. V., Bowyer, K. W., Hall, L. O., et al. "SMOTE: synthetic minority over-sampling technique". In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[38] Chen, C., Bai, W., Davies, R. H., et al. "Improving the generalizability of convolutional neural network-based segmentation on CMR images". In: *Frontiers in cardiovascular medicine* 7 (2020), p. 105.

[39] Chen, C., Li, Z., Ouyang, C., et al. "MaxStyle: Adversarial Style Composition for Robust Medical Image Segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI 2022)*. 2022.

[40] Chen, C., Ouyang, C., Li, Z., et al. "Enhancing MR Image Segmentation with Realistic Adversarial Data Augmentation". In: *Medical Image Analysis* (2022).

[41] Chen, C., Qin, C., Ouyang, C., et al. "Enhancing mr image segmentation with realistic adversarial data augmentation". In: *Medical Image Analysis* (2022), p. 102597.

[42] Chen, C., Qin, C., Qiu, H., et al. "Realistic adversarial data augmentation for MR image segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2020, pp. 667–677.

[43] Chen, J., Lu, Y., Yu, Q., et al. "Transunet: Transformers make strong encoders for medical image segmentation". In: *arXiv preprint arXiv:2102.04306* (2021).

[44] Chen, L.-C., Papandreou, G., Kokkinos, I., et al. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs". In: *IEEE transactions on pattern analysis and machine intelligence* 40.4 (2017), pp. 834–848.

[45] Chen, M., Goel, K., Sohoni, N. S., et al. "Mandoline: Model Evaluation under Distribution Shift". In: *ICML*. PMLR. 2021, pp. 1617–1629.

[46] Chongxuan, L., Xu, T., Zhu, J., et al. "Triple generative adversarial nets". In: *Advances in neural information processing systems*. 2017, pp. 4088–4098.

[47] Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., et al. "3D U-Net: learning dense volumetric segmentation from sparse annotation". In: *International conference on medical image computing and computer-assisted intervention*. Springer. 2016, pp. 424–432.

[48] Ciregan, D., Meier, U., and Schmidhuber, J. "Multi-column deep neural networks for image classification". In: *2012 IEEE conference on computer vision and pattern recognition*. IEEE. 2012, pp. 3642–3649.

[49] Cireşan, D. C., Giusti, A., Gambardella, L. M., et al. "Mitosis detection in breast cancer histology images with deep neural networks". In: *International Conference on Medical Image Computing and Computer-assisted Intervention*. Springer. 2013, pp. 411–418.

[50] Combalia, M., Codella, N. C., Rotemberg, V., et al. "Bcn20000: Dermoscopic lesions in the wild". In: *arXiv preprint arXiv:1908.02288* (2019).

[51] Cordts, M., Omran, M., Ramos, S., et al. "The cityscapes dataset for semantic urban scene understanding". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3213–3223.

[52] Cubuk, E. D., Zoph, B., Mane, D., et al. "Autoaugment: Learning augmentation policies from data". In: *arXiv preprint arXiv:1805.09501* (2018).

[53] Cubuk, E. D., Zoph, B., Mane, D., et al. "Autoaugment: Learning augmentation strategies from data". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2019, pp. 113–123.

[54] Cubuk, E. D., Zoph, B., Shlens, J., et al. "Randaugment: Practical automated data augmentation with a reduced search space". In: *arXiv preprint arXiv:1909.13719* (2019).

[55] Cui, Y., Jia, M., Lin, T.-Y., et al. "Class-balanced loss based on effective number of samples". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9268–9277.

[56] Deng, J., Guo, J., Xue, N., et al. "Arcface: Additive angular margin loss for deep face recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4690–4699.

[57] Deng, W. and Zheng, L. "Are labels always necessary for classifier accuracy evaluation?" In: *CVPR*. 2021, pp. 15069–15078.

[58] Devlin, J., Chang, M.-W., Lee, K., et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[59] Devlin, J., Zbib, R., Huang, Z., et al. "Fast and robust neural network joint models for statistical machine translation". In: *proceedings of the 52nd annual meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2014, pp. 1370–1380.

[60] DeVries, T. and Taylor, G. W. "Improved regularization of convolutional neural networks with cutout". In: *arXiv preprint arXiv:1708.04552* (2017).

[61] Dong, Q., Gong, S., and Zhu, X. "Imbalanced deep learning by minority class incremental rectification". In: *IEEE transactions on pattern analysis and machine intelligence* 41.6 (2018), pp. 1367–1381.

[62] Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[63] Dou, Q., Coelho de Castro, D., Kamnitsas, K., et al. "Domain generalization via model-agnostic learning of semantic features". In: *Advances in Neural Information Processing Systems* 32 (2019).

[64] Dou, Q., So, T. Y., Jiang, M., et al. "Federated deep learning for detecting COVID-19 lung abnormalities in CT: a privacy-preserving multinational validation study". In: *NPJ digital medicine* 4.1 (2021), pp. 1–11.

[65] Downey, A. B. "Evidence for long-tailed distributions in the internet". In: *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*. 2001, pp. 229–241.

[66] Eaton-Rosen, Z., Bragman, F., Ourselin, S., et al. "Improving data augmentation for medical image segmentation". In: *International Conference on Medical Imaging with Deep Learning* (2018).

[67] Eche, T., Schwartz, L. H., Mokrane, F.-Z., et al. "Toward Generalizability in the Deployment of Artificial Intelligence in Radiology: Role of Computation Stress Testing to Overcome Underspecification". In: *Radiology: Artificial Intelligence* 3.6 (2021), e210097.

[68] Elsahar, H. and Gallé, M. "To annotate or not? predicting performance drop under domain shift". In: *EMNLP-IJCNLP*. 2019, pp. 2163–2173.

[69] Esteva, A., Robicquet, A., Ramsundar, B., et al. "A guide to deep learning in healthcare". In: *Nature medicine* 25.1 (2019), pp. 24–29.

[70] Falkner, S., Klein, A., and Hutter, F. "BOHB: Robust and efficient hyperparameter optimization at scale". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 1437–1446.

[71] Fan, W. and Davidson, I. "Reverse testing: an efficient framework to select amongst classifiers under sample selection bias". In: *ACM SIGKDD*. 2006, pp. 147–156.

[72] Feichtenhofer, C., Pinz, A., Wildes, R. P., et al. "Deep insights into convolutional networks for video recognition". In: *International Journal of Computer Vision* 128.2 (2020), pp. 420–437.

[73] Finn, C., Abbeel, P., and Levine, S. "Model-agnostic meta-learning for fast adaptation of deep networks". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 1126–1135.

[74] Franceschi, L., Frasconi, P., Salzo, S., et al. "Bilevel programming for hyperparameter optimization and meta-learning". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 1568–1577.

[75] Frid-Adar, M., Klang, E., Amitai, M., et al. "Synthetic data augmentation using GAN for improved liver lesion classification". In: *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. IEEE. 2018, pp. 289–293.

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

[77] Garg, S., Balakrishnan, S., Lipton, Z. C., et al. "Leveraging unlabeled data to predict out-of-distribution performance". In: *ICLR*. 2022.

[78] Glocker, B., Pauly, O., Konukoglu, E., et al. "Joint classification-regression forests for spatially structured multi-object segmentation". In: *European conference on computer vision*. Springer. 2012, pp. 870–881.

[79] Goodfellow, I. J., Shlens, J., and Szegedy, C. "Explaining and harnessing adversarial examples". In: *arXiv preprint arXiv:1412.6572* (2014).

[80] Granese, F., Romanelli, M., Gorla, D., et al. "DOCTOR: A Simple Method for Detecting Misclassification Errors". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 5669–5681.

[81] Gu, X., Guo, Y., Li, Z., et al. "Tackling Long-Tailed Category Distribution Under Domain Shifts". In: *European Conference on Computer Vision (ECCV 2022)*. 2022.

[82] Guillory, D., Shankar, V., Ebrahimi, S., et al. "Predicting with confidence on unseen distributions". In: *ICCV*. 2021, pp. 1134–1144.

[83] Guo, C., Pleiss, G., Sun, Y., et al. "On calibration of modern neural networks". In: *ICML*. PMLR. 2017, pp. 1321–1330.

[84] Guo, H. and Viktor, H. L. "Learning from imbalanced data sets with boosting and data generation: the databoost-im approach". In: *ACM Sigkdd Explorations Newsletter* 6.1 (2004), pp. 30–39.

[85] Gupta, A., Venkatesh, S., Chopra, S., et al. "Generative Image Translation for Data Augmentation of Bone Lesion Pathology". In: *arXiv preprint arXiv:1902.02248* (2019).

[86] Gutman, D., Codella, N. C., Celebi, E., et al. "Skin lesion analysis toward melanoma detection: A challenge at the international symposium on biomedical imaging (ISBI) 2016, hosted by the international skin imaging collaboration (ISIC)". In: *arXiv preprint arXiv:1605.01397* (2016).

[87] Han, T., Xie, W., and Zisserman, A. "Self-supervised co-training for video representation learning". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 5679–5690.

[88] Hashemi, S. R., Salehi, S. S. M., Erdogmus, D., et al. "Asymmetric loss functions and deep densely-connected networks for highly-imbalanced medical image segmentation: Application to multiple sclerosis lesion detection". In: *IEEE Access* 7 (2018), pp. 1721–1735.

[89] Hataya, R., Zdenek, J., Yoshizoe, K., et al. "Faster autoaugment: Learning augmentation strategies using backpropagation". In: *European Conference on Computer Vision*. Springer. 2020, pp. 1–16.

[90] Hauberg, S., Freifeld, O., Larsen, A. B. L., et al. "Dreaming more data: Class-dependent distributions over diffeomorphisms for learned data augmentation". In: *Artificial Intelligence and Statistics*. 2016, pp. 342–350.

[91] Havaei, M., Davy, A., Warde-Farley, D., et al. "Brain tumor segmentation with deep neural networks". In: *Medical image analysis* 35 (2017), pp. 18–31.

[92] He, K., Fan, H., Wu, Y., et al. "Momentum contrast for unsupervised visual representation learning". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 9729–9738.

[93] He, K., Zhang, X., Ren, S., et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[94] He, K., Zhang, X., Ren, S., et al. "Identity mappings in deep residual networks". In: *European conference on computer vision*. Springer. 2016, pp. 630–645.

[95] He, Y., Carass, A., Zuo, L., et al. "Autoencoder based self-supervised test-time adaptation for medical image analysis". In: *Medical image analysis* 72 (2021), p. 102136.

[96] Heller, N., Sathianathen, N., Kalapara, A., et al. "The kits19 challenge data: 300 kidney tumor cases with clinical context, ct semantic segmentations, and surgical outcomes". In: *arXiv preprint arXiv:1904.00445* (2019).

[97] Hendrycks, D. and Dietterich, T. "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations". In: *Proceedings of the ICLR* (2019).

[98] Hendrycks, D. and Gimpel, K. "Gaussian error linear units (gelus)". In: *arXiv preprint arXiv:1606.08415* (2016).

[99] Hering, A., Hansen, L., Mok, T. C., et al. "Learn2Reg: comprehensive multi-task medical image registration challenge, dataset and evaluation in the era of deep learning". In: *arXiv preprint arXiv:2112.04489* (2021).

[100] Hinton, G., Deng, L., Yu, D., et al. "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups". In: *IEEE Signal processing magazine* 29.6 (2012), pp. 82–97.

[101] Holland, E. C. "Progenitor cells and glioma formation". In: *Current opinion in neurology* 14.6 (2001), pp. 683–688.

[102] Hospedales, T. M., Antoniou, A., Micaelli, P., et al. "Meta-learning in neural networks: A survey". In: *IEEE transactions on pattern analysis and machine intelligence* (2021).

[103] Houthooft, R., Chen, Y., Isola, P., et al. "Evolved policy gradients". In: *Advances in Neural Information Processing Systems* 31 (2018).

[104] Huang, C., Li, Y., Loy, C. C., et al. "Learning deep representation for imbalanced classification". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 5375–5384.

[105] Iglesias, J. E. and Sabuncu, M. R. "Multi-atlas segmentation of biomedical images: a survey". In: *Medical image analysis* 24.1 (2015), pp. 205–219.

[106] Isensee, F., Jaeger, P. F., Kohl, S. A., et al. "nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation". In: *Nature methods* 18.2 (2021), pp. 203–211.

[107] Isensee, F., Jäger, P. F., Kohl, S. A., et al. "Automated design of deep learning methods for biomedical image segmentation". In: *arXiv preprint arXiv:1904.08128* (2019).

[108] Islam, M. and Glocker, B. "Spatially Varying Label Smoothing: Capturing Uncertainty from Expert Annotations". In: *International Conference on Information Processing in Medical Imaging*. Springer. 2021, pp. 677–688.

[109] Islam, M., Seenivasan, L., Ren, H., et al. "Class-Distribution-Aware Calibration for Long-Tailed Visual Recognition". In: *arXiv preprint arXiv:2109.05263* (2021).

[110] Jang, E., Gu, S., and Poole, B. "Categorical reparameterization with gumbel-softmax". In: *arXiv preprint arXiv:1611.01144* (2016).

[111] Japkowicz, N. and Stephen, S. "The class imbalance problem: A systematic study". In: *Intelligent data analysis* 6.5 (2002), pp. 429–449.

[112] Jiang, H., Kim, B., Guan, M., et al. "To trust or not to trust a classifier". In: *Advances in neural information processing systems* 31 (2018).

[113] Jiao, J., Cai, Y., Alsharid, M., et al. "Self-supervised contrastive video-speech representation learning for ultrasound". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2020, pp. 534–543.

[114] Jin, D., Xu, Z., Tang, Y., et al. "CT-realistic lung nodule simulation from 3D conditional generative adversarial networks for robust lung segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2018, pp. 732–740.

[115] Johnson, J. M. and Khoshgoftaar, T. M. "Survey on deep learning with class imbalance". In: *Journal of Big Data* 6.1 (2019), pp. 1–54.

[116] Kamnitsas, K., Baumgartner, C., Ledig, C., et al. "Unsupervised domain adaptation in brain lesion segmentation with adversarial networks". In: *International conference on information processing in medical imaging*. Springer. 2017, pp. 597–609.

[117] Kamnitsas, K., Ledig, C., Newcombe, V. F., et al. "Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation". In: *Med. Image Anal.* 36 (2017), pp. 61–78.

[118] Kang, B., Xie, S., Rohrbach, M., et al. "Decoupling representation and classifier for long-tailed recognition". In: *International Conference on Learning Representations (ICLR)*. 2020.

[119] Karani, N., Erdil, E., Chaitanya, K., et al. "Test-time adaptable neural networks for robust medical image segmentation". In: *Medical Image Analysis* 68 (2021), p. 101907.

[120] Kawaguchi, K., Kaelbling, L. P., and Bengio, Y. "Generalization in deep learning". In: *arXiv preprint arXiv:1710.05468* (2017).

[121] Kawahara, J., Daneshvar, S., Argenziano, G., et al. "Seven-point checklist and skin lesion classification using multitask multimodal neural nets". In: *IEEE journal of biomedical and health informatics* 23.2 (2018), pp. 538–546.

[122] Kervadec, H., Bouchtiba, J., Desrosiers, C., et al. "Boundary loss for highly unbalanced segmentation". In: *International conference on medical imaging with deep learning*. 2019, pp. 285–296.

[123] Khan, S., Hayat, M., Zamir, S. W., et al. "Striking the right balance with uncertainty". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 103–112.

[124] Kim, I., Kim, Y., and Kim, S. "Learning loss for test-time augmentation". In: *arXiv preprint arXiv:2010.11422* (2020).

[125] Kingma, D. P. and Ba, J. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[126] Krizhevsky, A., Hinton, G., et al. "Learning multiple layers of features from tiny images". In: *Technical report* (2009).

[127] Krizhevsky, A., Sutskever, I., and Hinton, G. E. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[128] Krogh, A. and Hertz, J. A. "A simple weight decay can improve generalization". In: *Advances in neural information processing systems*. 1992, pp. 950–957.

[129] Krogh, A. and Vedelsby, J. "Neural network ensembles, cross validation, and active learning". In: *Advances in neural information processing systems*. 1995, pp. 231–238.

[130] Kumar, V., Gu, Y., Basu, S., et al. "Radiomics: the process and the challenges". In: *Magnetic resonance imaging* 30.9 (2012), pp. 1234–1248.

[131] Laine, S. and Aila, T. "Temporal ensembling for semi-supervised learning". In: *arXiv preprint arXiv:1610.02242* (2016).

[132] Landman, B., Xu, Z., Eugenio Igelsias, J., et al. *MICCAI Multi-Atlas Labeling Beyond the Cranial Vault–Workshop and Challenge*. 2015.

[133] LeCun, Y., Bengio, Y., and Hinton, G. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[134] LeCun, Y., Boser, B., Denker, J. S., et al. "Backpropagation applied to handwritten zip code recognition". In: *Neural computation* 1.4 (1989), pp. 541–551.

[135] LeCun, Y., Bottou, L., Bengio, Y., et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[136] Ledig, C., Heckemann, R. A., Hammers, A., et al. "Robust whole-brain segmentation: application to traumatic brain injury". In: *Medical image analysis* 21.1 (2015), pp. 40–58.

[137] Lemaître, G., Martí, R., Freixenet, J., et al. "Computer-aided detection and diagnosis for prostate cancer based on mono and multi-parametric MRI: a review". In: *Computers in biology and medicine* 60 (2015), pp. 8–31.

[138] Li, C., Xu, C., Gui, C., et al. "Level set evolution without re-initialization: a new variational formulation". In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Vol. 1. IEEE. 2005, pp. 430–436.

[139] Li, H., Han, H., Li, Z., et al. "High-Resolution Chest X-ray Bone Suppression Using Unpaired CT Structural Priors". In: *IEEE Transactions on Medical Imaging* (2020).

[140] Li, L., Jamieson, K., DeSalvo, G., et al. "Hyperband: A novel bandit-based approach to hyperparameter optimization". In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 6765–6816.

[141] Li, L., Ma, Q., Li, Z., et al. "Fetal Cortex Segmentation with Topology and Thickness Loss Constraints". In: *Topological Data Analysis and Its Applications for Medical Data (miccai-tda 2022)*. 2022.

[142] Li, X., Chen, H., Qi, X., et al. "H-DenseUNet: hybrid densely connected UNet for liver and tumor segmentation from CT volumes". In: *IEEE transactions on medical imaging* 37.12 (2018), pp. 2663–2674.

[143] Li, Y., Zaragoza, H., Herbrich, R., et al. "The perceptron algorithm with uneven margins". In: *ICML*. Vol. 2. 2002, pp. 379–386.

[144] Li, Y., Yang, Y., Zhou, W., et al. "Feature-critic networks for heterogeneous domain generalization". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 3915–3924.

[145] Li, Y., Hu, G., Wang, Y., et al. "Dada: Differentiable automatic data augmentation". In: *arXiv preprint arXiv:2003.03780* (2020).

[146] Li, Z., Kamnitsas, K., and Glocker, B. "Overfitting of neural nets under class imbalance: Analysis and improvements for segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2019, pp. 402–410.

[147] Li, Z., Kamnitsas, K., and Glocker, B. "Analyzing overfitting under class imbalance in neural networks for image segmentation". In: *IEEE Transactions on Medical Imaging* 40.3 (2020), pp. 1065–1077.

[148] Li, Z., Kamnitsas, K., Islam, M., et al. "Estimating Model Performance under Domain Shifts with Class-Specific Confidence Scores". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI 2022)*. 2022.

[149] Li, Z., Kamnitsas, K., Ouyang, C., et al. "Context Label Learning: Improving Background Class Representations in Semantic Segmentation". In: *arXiv preprint arXiv:2212.08423* (2022).

[150] Li, Z., Li, H., Han, H., et al. "Encoding CT Anatomy Knowledge for Unpaired Chest X-ray Image Decomposition". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI 2019)*. 2019.

[151] Li, Z., Yu, J., Wang, Y., et al. "DeepVolume: Brain Structure and Spatial Connection-Aware Network for Brain MRI Super-Resolution". In: *IEEE transactions on cybernetics* (2019).

[152] Liew, S.-L., Anglin, J. M., Banks, N. W., et al. "A large, open source dataset of stroke anatomical brain images and manual lesion segmentations". In: *Scientific data* 5 (2018), p. 180011.

[153] Lim, S., Kim, I., Kim, T., et al. "Fast AutoAugment". In: *arXiv preprint arXiv:1905.00397* (2019).

[154] Lin, T.-Y., Goyal, P., Girshick, R., et al. "Focal loss for dense object detection". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.

[155] Litjens, G., Kooi, T., Bejnordi, B. E., et al. "A survey on deep learning in medical image analysis". In: *Medical image analysis* 42 (2017), pp. 60–88.

[156] Litjens, G., Toth, R., Ven, W. van de, et al. "Evaluation of prostate segmentation algorithms for MRI: the PROMISE12 challenge". In: *Medical image analysis* 18.2 (2014), pp. 359–373.

[157] Liu, H., Simonyan, K., and Yang, Y. "Darts: Differentiable architecture search". In: *arXiv preprint arXiv:1806.09055* (2018).

[158] Liu, Q., Dou, Q., and Heng, P. A. "Shape-aware Meta-learning for Generalizing Prostate MRI Segmentation to Unseen Domains". In: *MICCAI*. 2020.

[159] Liu, Q., Dou, Q., and Heng, P.-A. "Shape-aware meta-learning for generalizing prostate MRI segmentation to unseen domains". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2020, pp. 475–485.

[160] Liu, S., Davison, A. J., and Johns, E. "Self-supervised generalisation with meta auxiliary learning". In: *arXiv preprint arXiv:1901.08933* (2019).

[161] Liu, W., Wen, Y., Yu, Z., et al. "Large-Margin Softmax Loss for Convolutional Neural Networks". In: *International Conference on Machine Leanring (ICML)*. 2016, pp. 507–516.

[162] Liu, Y., Albanie, S., Nagrani, A., et al. "Use what you have: Video retrieval using representations from collaborative experts". In: *arXiv preprint arXiv:1907.13487* (2019).

[163] Liu, Z., Mao, H., Wu, C.-Y., et al. "A convnet for the 2020s". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 11976–11986.

[164] Liu, Z., Miao, Z., Zhan, X., et al. "Large-scale long-tailed recognition in an open world". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2537–2546.

[165] Long, J., Shelhamer, E., and Darrell, T. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[166] Luo, H., Zhuang, Q., Wang, Y., et al. "A novel image signature-based radiomics method to achieve precise diagnosis and prognostic stratification of gliomas". In: *Laboratory investigation* 101.4 (2021), pp. 450–462.

[167] Maas, A. L., Hannun, A. Y., Ng, A. Y., et al. "Rectifier nonlinearities improve neural network acoustic models". In: *Proc. icml*. Vol. 30. 1. Citeseer. 2013, p. 3.

[168] Marrakchi, Y., Makansi, O., and Brox, T. "Fighting Class Imbalance with Contrastive Learning". In: *MICCAI*. Springer. 2021, pp. 466–476.

[169] McInerney, T. and Terzopoulos, D. "Deformable models in medical image analysis: a survey". In: *Medical image analysis* 1.2 (1996), pp. 91–108.

[170] Mehrabi, N., Morstatter, F., Saxena, N., et al. "A survey on bias and fairness in machine learning". In: *ACM Computing Surveys (CSUR)* 54.6 (2021), pp. 1–35.

[171] Mehrtash, A., Wells, W. M., Tempany, C. M., et al. "Confidence calibration and predictive uncertainty estimation for deep medical image segmentation". In: *IEEE transactions on medical imaging* 39.12 (2020), pp. 3868–3878.

[172] Mendonça, T., Ferreira, P. M., Marques, J. S., et al. "PH 2-A dermoscopic image database for research and benchmarking". In: *EMBC*. IEEE. 2013, pp. 5437–5440.

[173] Menze, B. H., Jakab, A., Bauer, S., et al. "The multimodal brain tumor image segmentation benchmark (BRATS)". In: *IEEE transactions on medical imaging* 34.10 (2014), pp. 1993–2024.

[174] Metz, L., Maheswaranathan, N., Cheung, B., et al. "Meta-learning update rules for unsupervised representation learning". In: *arXiv preprint arXiv:1804.00222* (2018).

[175] Milletari, F., Navab, N., and Ahmadi, S.-A. "V-net: Fully convolutional neural networks for volumetric medical image segmentation". In: *2016 Fourth International Conference on 3D Vision (3DV)*. IEEE. 2016, pp. 565–571.

[176] Minaee, S., Boykov, Y. Y., Porikli, F., et al. "Image segmentation using deep learning: A survey". In: *IEEE transactions on pattern analysis and machine intelligence* (2021).

[177] Miyato, T., Maeda, S.-i., Ishii, S., et al. "Virtual adversarial training: a regularization method for supervised and semi-supervised learning". In: *IEEE transactions on pattern analysis and machine intelligence* (2018).

[178] Moeskops, P., Viergever, M. A., Mendrik, A. M., et al. "Automatic segmentation of MR brain images with a convolutional neural network". In: *IEEE transactions on medical imaging* 35.5 (2016), pp. 1252–1261.

[179] Müller, R., Kornblith, S., and Hinton, G. "When does label smoothing help?" In: *arXiv preprint arXiv:1906.02629* (2019).

[180] Nakkiran, P., Kaplun, G., Bansal, Y., et al. "Deep double descent: Where bigger models and more data hurt". In: *Journal of Statistical Mechanics: Theory and Experiment* 2021.12 (2021), p. 124003.

[181] Namburete, A. I., Xie, W., Yaqub, M., et al. "Fully-automated alignment of 3D fetal brain ultrasound to a canonical reference space using multi-task learning". In: *Medical image analysis* 46 (2018), pp. 1–14.

[182] Navon, A., Achituve, I., Maron, H., et al. "Auxiliary learning by implicit differentiation". In: *arXiv preprint arXiv:2007.02693* (2020).

[183] Neelakantan, A., Vilnis, L., Le, Q. V., et al. "Adding gradient noise improves learning for very deep networks". In: *arXiv preprint arXiv:1511.06807* (2015).

[184] Neyshabur, B., Bhojanapalli, S., McAllester, D., et al. "Exploring generalization in deep learning". In: *Advances in Neural Information Processing Systems*. 2017, pp. 5947–5956.

[185] Novak, R., Bahri, Y., Abolafia, D. A., et al. "Sensitivity and generalization in neural networks: an empirical study". In: *arXiv preprint arXiv:1802.08760* (2018).

[186] Oktay, O., Schlemper, J., Folgoc, L. L., et al. "Attention u-net: Learning where to look for the pancreas". In: *arXiv preprint arXiv:1804.03999* (2018).

[187] Ouyang, C., Chen, C., Li, S., et al. "Causality-inspired Single-source Domain Generalization for Medical Image Segmentation". In: *arXiv preprint arXiv:2111.12525* (2021).

[188] Ouyang, C., Wang, S., Chen, C., et al. "Improved post-hoc probability calibration for artifact-corrupted MRI segmentation". In: *Uncertainty for Safe Utilization of Machine Learning in Medical Imaging (miccai-unsure 2022)*. 2022.

[189] Pan, T.-Y., Zhang, C., Li, Y., et al. "On model calibration for long-tailed object detection and instance segmentation". In: *NeurIPS* 34 (2021).

[190] Paschali, M., Simson, W., Roy, A. G., et al. "Data augmentation with manifold exploring geometric transformations for increased performance and robustness". In: *arXiv preprint arXiv:1901.04420* (2019).

[191] Paszke, A., Gross, S., Massa, F., et al. "Pytorch: An imperative style, high-performance deep learning library". In: *Advances in neural information processing systems* 32 (2019), pp. 8026–8037.

[192] Pedregosa, F. "Hyperparameter optimization with approximate gradient". In: *International conference on machine learning*. PMLR. 2016, pp. 737–746.

[193] Peng, X., Tang, Z., Yang, F., et al. "Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2226–2234.

[194] Pham, D. L., Xu, C., and Prince, J. L. "A survey of current methods in medical image segmentation". In: *Annual review of biomedical engineering* 2.3 (2000), pp. 315–337.

[195] Poggio, T., Kawaguchi, K., Liao, Q., et al. "Theory of Deep Learning III: explaining the non-overfitting puzzle". In: *arXiv preprint arXiv:1801.00173* (2017).

[196] Qaiser, T., Winzeck, S., Barfoot, T., et al. "Multiple Instance Learning with Auxiliary Task Weighting for Multiple Myeloma Classification". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2021, pp. 786–796.

[197] Qiao, M., Liu, C., Li, Z., et al. "Breast Tumor Classification based on MRI-US Images by Disentangling Modality Features". In: *IEEE Journal of Biomedical and Health Informatics* (2022).

[198] Radford, A., Kim, J. W., Hallacy, C., et al. "Learning transferable visual models from natural language supervision". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8748–8763.

[199] Ramesh, A., Pavlov, M., Goh, G., et al. "Zero-shot text-to-image generation". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8821–8831.

[200] Ratner, A. J., Ehrenberg, H., Hussain, Z., et al. "Learning to compose domain-specific transformations for data augmentation". In: *Advances in neural information processing systems*. 2017, pp. 3236–3246.

[201] Ren, J., Zhang, M., Yu, C., et al. "Balanced MSE for Imbalanced Visual Regression". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 7926–7935.

[202] Robbins, H. and Monro, S. "A stochastic approximation method". In: *The annals of mathematical statistics* (1951), pp. 400–407.

[203] Robinson, R., Oktay, O., Bai, W., et al. "Real-time prediction of segmentation quality". In: *MICCAI*. Springer. 2018, pp. 578–585.

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

[205] Rotemberg, V., Kurtansky, N., Betz-Stablein, B., et al. "A patient-centric dataset of images and metadata for identifying melanomas using clinical context". In: *Scientific data* 8.1 (2021), pp. 1–8.

[206] Rueckert, D. and Schnabel, J. A. "Model-based and data-driven strategies in medical image computing". In: *Proceedings of the IEEE* 108.1 (2019), pp. 110–124.

[207] Russakovsky, O., Deng, J., Su, H., et al. "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115.3 (2015), pp. 211–252.

[208] Savenije, M. H., Maspero, M., Sikkes, G. G., et al. "Clinical implementation of MRI-based organs-at-risk auto-segmentation with convolutional networks for prostate radiotherapy". In: *Radiation Oncology* 15 (2020), pp. 1–12.

[209] Schmidhuber, J. "Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook". PhD thesis. Technische Universität München, 1987.

[210] Setio, A. A. A., Ciompi, F., Litjens, G., et al. "Pulmonary nodule detection in CT images: false positive reduction using multi-view convolutional networks". In: *IEEE Trans. Med. Imaging* 35.5 (2016), pp. 1160–1169.

[211] Shanmugam, D., Blalock, D., Balakrishnan, G., et al. "When and why test-time augmentation works". In: *arXiv preprint arXiv:2011.11156* (2020).

[212] Shapey, J., Wang, G., Dorent, R., et al. "An artificial intelligence framework for automatic segmentation and volumetry of vestibular schwannomas from contrast-enhanced T1-weighted and high-resolution T2-weighted MRI". In: *Journal of neurosurgery* 134.1 (2019), pp. 171–179.

[213] Sharma, H., Drukker, L., Chatelain, P., et al. "Knowledge representation and learning of operator clinical workflow from full-length routine fetal ultrasound scan videos". In: *Medical Image Analysis* 69 (2021), p. 101973.

[214] Shelmerdine, S. C., Martin, H., Shirodkar, K., et al. "Can artificial intelligence pass the Fellowship of the Royal College of Radiologists examination? Multi-reader diagnostic accuracy study". In: *bmj* 379 (2022).

[215] Shen, W., Shi, Z., and Sun, J. "Learning from Adversarial Features for Few-Shot Classification". In: *arXiv preprint arXiv:1903.10225* (2019).

[216] Shorten, C. and Khoshgoftaar, T. M. "A survey on image data augmentation for deep learning". In: *Journal of big data* 6.1 (2019), pp. 1–48.

[217] Silver, D., Huang, A., Maddison, C. J., et al. "Mastering the game of Go with deep neural networks and tree search". In: *nature* 529.7587 (2016), pp. 484–489.

[218] Simonyan, K. and Zisserman, A. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[219] Simpson, A. L., Antonelli, M., Bakas, S., et al. "A large annotated medical image dataset for the development and evaluation of segmentation algorithms". In: *arXiv preprint arXiv:1902.09063* (2019).

[220] Snell, J., Swersky, K., and Zemel, R. "Prototypical networks for few-shot learning". In: *Advances in neural information processing systems* 30 (2017).

[221] Srebro, N. and Shraibman, A. "Rank, trace-norm and max-norm". In: *International Conference on Computational Learning Theory*. Springer. 2005, pp. 545–560.

[222] Srivastava, N., Hinton, G., Krizhevsky, A., et al. "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[223] Stanley, K. O., Clune, J., Lehman, J., et al. "Designing neural networks through neuroevolution". In: *Nature Machine Intelligence* 1.1 (2019), pp. 24–35.

[224] Sun, C., Shrivastava, A., Singh, S., et al. "Revisiting unreasonable effectiveness of data in deep learning era". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 843–852.

[225] Sutskever, I., Vinyals, O., and Le, Q. V. "Sequence to sequence learning with neural networks". In: *Advances in neural information processing systems* 27 (2014).

[226] Tang, K., Huang, J., and Zhang, H. "Long-tailed classification by keeping the good and removing the bad momentum causal effect". In: *arXiv preprint arXiv:2009.12991* (2020).

[227] Tran, T., Pham, T., Carneiro, G., et al. "A Bayesian data augmentation approach for learning deep models". In: *Advances in Neural Information Processing Systems*. 2017, pp. 2797–2806.

[228] Tschandl, P., Rosendahl, C., and Kittler, H. "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions". In: *Scientific data* 5.1 (2018), pp. 1–9.

[229] Tzeng, E., Hoffman, J., Saenko, K., et al. "Adversarial discriminative domain adaptation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7167–7176.

[230] Valindria, V. V., Lavdas, I., Bai, W., et al. "Reverse classification accuracy: predicting segmentation performance in the absence of ground truth". In: *IEEE transactions on medical imaging* 36.8 (2017), pp. 1597–1606.

[231] Valindria, V. V., Lavdas, I., Cerrolaza, J., et al. "Small organ segmentation in whole-body MRI using a two-stage FCN and weighting schemes". In: *International Workshop on Machine Learning in Medical Imaging*. Springer. 2018, pp. 346–354.

[232] Van Amersfoort, J., Smith, L., Teh, Y. W., et al. "Uncertainty estimation using a single deep deterministic neural network". In: *International conference on machine learning*. PMLR. 2020, pp. 9690–9700.

[233] Van Dyk, D. A. and Meng, X.-L. "The art of data augmentation". In: *Journal of Computational and Graphical Statistics* 10.1 (2001), pp. 1–50.

[234] Vapnik, V. N. "An overview of statistical learning theory". In: *IEEE transactions on neural networks* 10.5 (1999), pp. 988–999.

[235] Wang, F., Cheng, J., Liu, W., et al. "Additive margin softmax for face verification". In: *IEEE Signal Processing Letters* 25.7 (2018), pp. 926–930.

[236] Wang, G., Li, W., Aertsen, M., et al. "Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks". In: *Neurocomputing* 338 (2019), pp. 34–45.

[237] Wang, S., Liu, W., Wu, J., et al. "Training deep neural networks on imbalanced data sets". In: *2016 international joint conference on neural networks (IJCNN)*. IEEE. 2016, pp. 4368–4374.

[238] Wang, Y.-X., Ramanan, D., and Hebert, M. "Learning to model the tail". In: *Advances in Neural Information Processing Systems*. 2017, pp. 7029–7039.

[239] Wernick, M. N., Yang, Y., Brankov, J. G., et al. "Machine learning in medical imaging". In: *IEEE signal processing magazine* 27.4 (2010), pp. 25–38.

[240] Wong, K. C., Moradi, M., Tang, H., et al. "3D segmentation with exponential logarithmic loss for highly unbalanced object sizes". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2018, pp. 612–619.

[241] Woolrich, M. W., Behrens, T. E. J., Beckmann, C. F., et al. "Mixture models with adaptive spatial regularization for segmentation with an application to fMRI data". In: *IEEE transactions on medical imaging* 24.1 (2005), pp. 1–11.

[242] Wright, J., Yang, A. Y., Ganesh, A., et al. "Robust face recognition via sparse representation". In: *IEEE transactions on pattern analysis and machine intelligence* 31.2 (2008), pp. 210–227.

[243] Xiang, L., Ding, G., and Han, J. "Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification". In: *European Conference on Computer Vision*. Springer. 2020, pp. 247–263.

[244] Xie, C., Tan, M., Gong, B., et al. "Adversarial Examples Improve Image Recognition". In: *arXiv preprint arXiv:1911.09665* (2019).

[245] Xie, Q., Dai, Z., Hovy, E., et al. "Unsupervised data augmentation". In: *arXiv preprint arXiv:1904.12848* (2019).

[246] Xu, C. and Prince, J. L. "Snakes, shapes, and gradient vector flow". In: *IEEE Transactions on image processing* 7.3 (1998), pp. 359–369.

[247] Xu, J., Li, M., and Zhu, Z. "Automatic data augmentation for 3D medical image segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2020, pp. 378–387.

[248] Yang, D., Roth, H., Xu, Z., et al. "Searching learning strategy with reinforcement learning for 3d medical image segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2019, pp. 3–11.

[249] Yifan, W., Doersch, C., Arandjelović, R., et al. "Input-level Inductive Biases for 3D Reconstruction". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 6176–6186.

[250] Yoon, C., Hamarneh, G., and Garbi, R. "Generalizable feature learning in the presence of data bias and domain class imbalance with application to skin lesion classification". In: *MICCAI*. Springer. 2019, pp. 365–373.

[251] Yoshida, Y. and Miyato, T. "Spectral norm regularization for improving the generalizability of deep learning". In: *arXiv preprint arXiv:1705.10941* (2017).

[252] Yun, S., Han, D., Oh, S. J., et al. "Cutmix: Regularization strategy to train strong classifiers with localizable features". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 6023–6032.

[253] Zadrozny, B., Langford, J., and Abe, N. "Cost-sensitive learning by cost-proportionate example weighting". In: *Third IEEE international conference on data mining*. IEEE. 2003, pp. 435–442.

[254] Zamir, A. R., Sax, A., Shen, W., et al. "Taskonomy: Disentangling task transfer learning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 3712–3722.

[255] Zela, A., Elsken, T., Saikia, T., et al. "Understanding and robustifying differentiable architecture search". In: *arXiv preprint arXiv:1909.09656* (2019).

[256] Zhang, C., Bengio, S., Hardt, M., et al. "Understanding deep learning requires rethinking generalization". In: *International Conference on Learning Representations (ICLR)*. 2017.

[257] Zhang, C., Bengio, S., Hardt, M., et al. "Understanding deep learning (still) requires rethinking generalization". In: *Communications of the ACM* 64.3 (2021), pp. 107–115.

[258] Zhang, D., Huang, G., Zhang, Q., et al. "Exploring task structure for brain tumor segmentation from multi-modality MR images". In: *IEEE Transactions on Image Processing* 29 (2020), pp. 9032–9043.

[259] Zhang, D., Zhang, J., Zhang, Q., et al. "Automatic pancreas segmentation based on lightweight DCNN modules and spatial prior propagation". In: *Pattern Recognition* 114 (2021), p. 107762.

[260] Zhang, D., Zhang, H., Tang, J., et al. "Causal intervention for weakly-supervised semantic segmentation". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 655–666.

[261] Zhang, H., Cisse, M., Dauphin, Y. N., et al. "mixup: Beyond empirical risk minimization". In: *arXiv preprint arXiv:1710.09412* (2017).

[262] Zhang, Y., Yang, L., Zheng, H., et al. "SPDA: Superpixel-based Data Augmentation for Biomedical Image Segmentation". In: *arXiv preprint arXiv:1903.00035* (2019).

[263] Zhao, A., Balakrishnan, G., Durand, F., et al. "Data augmentation using learned transforms for one-shot medical image segmentation". In: *arXiv preprint arXiv:1902.09383* (2019).

[264] Zheng, W., Chen, Z., Lu, J., et al. "Hardness-Aware Deep Metric Learning". In: *arXiv preprint arXiv:1903.05503* (2019).

[265] Zhou, B., Cui, Q., Wei, X.-S., et al. "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.

[266] Zhou, H., Yuan, Y., and Shi, C. "Object tracking using SIFT features and mean shift". In: *Computer vision and image understanding* 113.3 (2009), pp. 345–352.

[267] Zhou, K., Liu, Z., Qiao, Y., et al. "Domain generalization: A survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[268] Zhou, P. and Feng, J. "Understanding Generalization and Optimization Performance of Deep CNNs". In: *International Conference on Machine Learning*. 2018.

[269] Zhou, Y., Li, Z., Bai, S., et al. "Prior-aware neural network for partially-supervised multi-organ segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 10672–10681.

[270] Zhou, Y., Xie, L., Shen, W., et al. "A fixed-point model for pancreas segmentation in abdominal CT scans". In: *International conference on medical image computing and computer-assisted intervention*. Springer. 2017, pp. 693–701.

[271] Zhou, Z., Rahman Siddiquee, M. M., Tajbakhsh, N., et al. "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.

[272] Zoph, B. and Le, Q. V. "Neural architecture search with reinforcement learning". In: *International Conference on Learning Representations (ICLR)*. 2017.