THE AUSTRALIAN NATIONAL UNIVERSITY

DOCTORAL THESIS

Label Shift Problem in Image Classification Tasks – A Comprehensive Analysis

Author:
Changkun Ye

Supervisors:
Prof. Nick Barnes,
Dr. Lars Petersson,
Dr. Russell Tsuchida

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

ANU College of Computing
ANU College of Engineering, Computing and Cybernetics

20 July 2025

Declaration

This dissertation is an account of research undertaken between Aug 2018 and Dec 2024 at the College of Computing, The Australian National University, Canberra, Australia. The dissertation has not been submitted to obtain a degree or diploma at any other university. To the best of my knowledge, it does not contain material previously published by another person, except where due reference is made.

Changkun Ye
20 July 2025

Acknowledgements

First and foremost, I would like to thank my supervisors Nick Barnes, Lars Petersson and Russell Tsuchida, for their support, guidance, and invaluable insights throughout the past years of my PhD candidature. Especially, I would like to thank my supervisor Nick Barnes for his patience and support during the early years of my PhD journey, thank my supervisor Lars Petersson for his valuable suggestions on the research directions and thank my supervisor Russell Tsuchida for his guidance and expertise that helps shape my works. One cannot ask supervisors better than that. I look forward to work with them in the future.

I would also like to thank my friends in the College of computing at the Australian National University and the colleagues in the Black Mountain site CISRO. Thank you for the valuable academic discussions that we had and and thank you for sharing the knowledge and experience in all the seminars and reading groups.

More generally, I would like to acknowledge the computer vision and machine learning communities for making these areas of research so dynamic and attractive. My thanks to those conference and journal reviewers whose feedback and suggestions have significantly improved my research.

I also want to acknowledge the various sources of funding that have assisted me during my PhD studies. This research could not be made possible without the financial support of the Data61 PhD scholarship offered by CSIRO and the research funding provided by the Australian National University.

Finally I would like to thank my family for their patience and support during my entire academic journey, especially my mum, who help me to establish confidence to pursue this PhD degree.

Abstract

Keywords: Distribution Shift, Label Shift, Class Imbalance Problem, EM algorithm

The real world deployment of machine learning models usually faces problems of distribution shift. Label shift is a common type of distribution shift that happens in classification tasks, where the source domain (train set) and target domain (test set) have different label distributions $p(y)$ but identical distributions of image given label $p(x|y)$. Under label shift, the classifier trained on the source domain may not perform as well in target domain.

Three problems arise under label shift: 1) *detection*: detect if label shift occurs, 2) *estimation*: estimate the target label distribution when no target domain labels are available and 3) *correction*: adapt a source domain classifier to the target domain under label shift. In practical problems, a label shift estimation model is first deployed to obtain an estimate of the target domain label distribution. A label shift correction model is then used to obtain the desired target domain classifier based on the estimate of the target label distribution.

This thesis investigates the label shift problems in the closed set classification, open set classification and zero-shot classification tasks.

In the closed set classification setting, a novel classifier based label shift estimation model and a more general feature based label shift estimation model are proposed. Based on reasonable assumptions, the two proposed models estimate label shift through EM algorithms, which are proved to converge to a Maximum Likelihood Estimate or Maximum *a Posteriori* estimate of the target label distribution. The estimation results are then used in a label shift correction model to obtain a target domain classifier without retraining or fine-tuning the original source domain classifier.

In the open set classification setting, a novel classifier based label shift estimation and correction model is proposed. With the help of a reference out-of-distribution dataset at test time, the proposed model estimates the target domain label distribution for both in-distribution classes that appear in the source domain as well as the extra out-of-distribution class. Under reasonable assumptions, the proposed model is proved to converge to a Maximum Likelihood Estimate of the target domain label distribution. The estimation result is then used in a label shift correction model to obtain a target domain classifier based on a pre-trained source domain in-distribution class classifier and a out-of-distribution binary classifier without re-training or fine-tuning.

In the zero-shot learning setting, a label shift correction model is proposed. A novel class-balanced triplet loss is proposed to help cluster features of each class equally when the source domain has an imbalanced label distribution. A Gaussian Process Regression model is then proposed to predict feature prototypes of unseen classes based on feature prototypes of seen classes. The final zero-shot learning classifier is then constructed, which is robust to source domain label shift.

Contents

Figures

Tables

Abbreviations

Abbreviations	Original
ANU	**A**ustralian **N**ational **U**niversity
CSIRO	**C**ommonwealth **S**cientific and **I**ndustrial **R**esearch **O**rganisation
LS	Label Shift
CSLS	Closed Set Label Shift
OSLS	Open Set Label Shift
GP	Gaussian Process
GPR	Gaussian Process Regression
KRR	Kernel Ridge Regression
LVM	Latent Variable Model
MM	Mixture Model
GMM	Gaussian Mixture Model
EM	Expectation Maximization
ZSL	Zero-Shot Learning
GZSL	Generalized Zero-Shot Learning
ID	In-Distribution
OOD	Out-of-Distribution
RV	Random Variable
i.i.d.	Independent and identically distributed
NLL	Negative Log Likelihood
MLE	Maximum Likelihood estimate
MAP	Maximum *a Posteriori*
MCMC	Markov chain Monte Carlo
NN	Neural Network
CNN	Convolutional Neural Network
DNN	Deep Neural Network
FC	Fully Connected
SGD	Stochastic Gradient Descent
SOTA	State-of-the-art

Top1 Acc	Top-one Accuracy
MSE	Mean Square Error
CE	Cross Entropy
t-SNE	t-distributed Stochastic Neighbor Embedding
AUROC	Area Under the Receiver Operating Characteristic curve
AUSUC	Area Under Seen and Unseen Curve
w.r.t.	With Respect To

List of Symbols

Object	Notation	Example(s)	Example Explanation
Set (General)	Calligraphic	$\mathcal{X}$	$\mathcal{X} \subseteq \mathbb{R}^K$
Real numbers set	$\mathbb{R}$	$\mathbb{R}^K_{>0}$	K dimensional positive real vectors
Probability simplex set	Δ	Δ^{K-1}	K dimensional probability simplexes
Random Variable (RV)	Capital	X	RV of data
Distribution of RV	$p(\cdot)$	$p(x)$	RV $X \sim p(x)$ follows distribution $p(x)$.
Realisation of RV	Lower case	x_0	$x_0 \sim_{i.i.d.} p(x)$
Constant	$Const$,C	-	-
L^p norm	$\|\cdot\|_p$	$\|\boldsymbol{x}\|_2$	L2-norm of vector $\boldsymbol{x}$
Indicator function	$\mathbb{I}(\cdot)$	$\mathbb{I}_y(x)$	$\mathbb{I}_y(x) = \begin{cases} 1, & x = y \\ 0, & \text{otherwise} \end{cases}$
Source domain RV	Subscript s	X_s, Y_s	Source domain data and label RV
Target domain RV	Subscript t	X_t, Y_t	Target domain data and label RV
Source domain distribution	$p_s(\cdot)$	$p_s(y)$	Source domain label distribution
Target domain distribution	$p_t(\cdot)$	$p_s(x)$	Target domain data distribution
Categorical distribution	$\text{Cat}(\cdot,\cdot)$	$\text{Cat}(K, \boldsymbol{c})$	K-dimensional categorical distribution with parameter $\boldsymbol{c} \in \Delta^{K-1}$
Dirichlet distribution	$\text{Dir}(\cdot,\cdot)$	$\text{Dir}(K, \boldsymbol{\alpha})$	K-dimensional Dirichlet distribution with parameter $\boldsymbol{\alpha} \in \mathbb{R}^K_{>0}$

Beta distribution	$\mathrm{Beta}(\cdot,\cdot)$	$\mathrm{Beta}(\alpha_1,\alpha_2)$	Beta distribution with parameter $\alpha_1,\alpha_2 \in \mathbb{R}_{>0}$

To my family

Chapter 1

Introduction

In recent years, research in image classification tasks has significantly advanced with the help of the Neural Network (NN) models. Deep Neural Network (DNN) models, which learn from labeled images, can exhibit human-like performance in image classification tasks when the test data is within the training categories.

Despite the outstanding performance of the Neural Network models on the image datasets, real world application of these models can sometimes yield unsatisfactory performance due to the difference between the training data distribution and the test data distribution. Thus, adapting a machine learning model under distribution shift has become an active research area.

This thesis focuses on a particular type of distribution shift – label shift. The problem of label shift attracts research interest due to its potential usage in classification tasks like Face Recognition (Huang, Li, Loy et al., 2019a), Medical Diagnosis (Suresh et al., 2023) as well as classification tasks in Satellite Imagery (Koßmann et al., 2021) and Genomic Applications (Yoon and Kwek, 2005), *etc*.

1.1 Label Shift Problem

The label shift problem can be illustrated with a simple example. Suppose we are training an image classifier to recognize animals commonly found in a group of nearby cities (*e.g.*, City A, City B, *etc*.) within the same region. Since these cities are geographically close, the types of animals seen are the same across them (*e.g.*, dogs, cats, and birds), and their appearances are also similar. For instance, an image of a dog captured in City A is just as likely to resemble a dog from City B.

During training, to reduce the effort of gathering and labeling images from all cities, we only use labeled images from City A, where dog images are more frequent than those of cats or birds. Due to this imbalance in the label distribution, the classifier trained on City A's dataset is biased towards the "dog" category (Buda et al., 2018a).

At test time, the classifier is deployed across all the cities. Take City B as an example — here, in contrast to City A, birds are more commonly seen than dogs or cats. Consequently, the classifier from City A that favors the dog category will perform sub-optimally in City B, even though the appearances of the animals are similar in both cities. The discrepancy between the label distributions in City A and City B

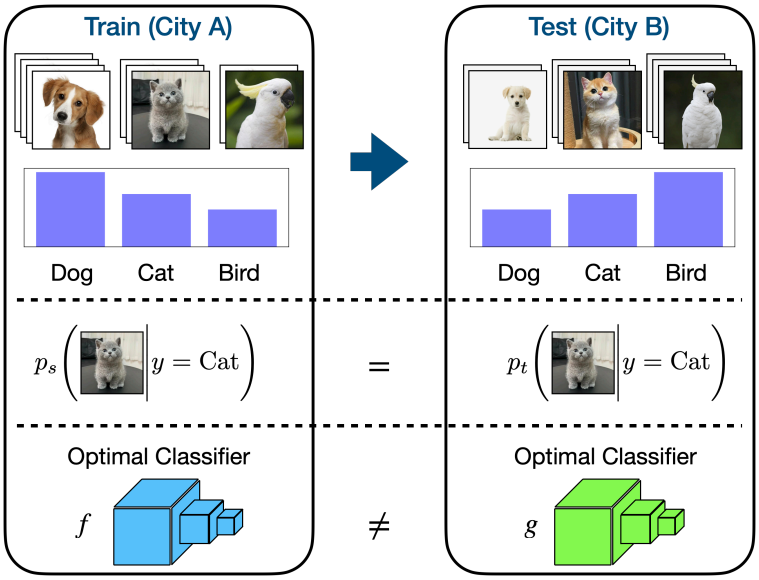

Figure 1.1: Example of the label shift problem. When the source domain (train set) and the target domain (test set) have different label distributions $p_s(y) \neq p_t(y)$. Even if we assume the two domains have identical conditional distribution of image given labels $p_s(x|y) = p_t(x|y)$, the optimal classifier f on the source domain may no longer be optimal on the target domain.

means that a classifier optimized for City A may not generalize well to City B, and *vice versa* (Lipton et al., 2018).

Such a problem is referred to as the Label Shift Problem. To address this issue, we should adjust the classifier when deploying it in other cities. Specifically, we want to adapt the classifier trained on the dataset collected from the data distribution in City A to the data distribution of City B, given that we have labeled images from City A but only unlabeled images from City B. Achieving this involves tackling three key sub-problems:

1. **Detection**: Verify that City B has a different label distribution compared with City A. If the label distributions between the two cities are the same, deploying the classifier in City B requires no further adjustment.
2. **Estimation**: If the label distributions differ between City A and City B, estimate the label distribution in City B using the labeled dataset from City A, the unlabeled dataset from City B, and the trained classifier.
3. **Correction**: Based on the estimate of the label distribution in City B and data from City A and City B, adjust the classifier trained on City A to suit City B.

These three tasks are thus referred to as the label shift *detection*, *estimation* and *correction* problems in the label shift literature (Alexandari et al., 2020; Lipton et al., 2018). Such problems are referred as the Closed Set Label Shift (CSLS) problem in this thesis, as the source and target domain has identical set of classes. The detailed mathematical definition of these problems will be discussed in the next chapter (Section 2.1). In practical problems, a label shift estimation model is first deployed to obtain an estimate of the target domain label distribution. A label shift correction

model is then used to get the desired target domain classifier based on the estimate of the target label distribution.

When City B has an extra unique class (*e.g.*zebra in the zoo) compared with City A, the label shift problem becomes the Open Set Label Shift (OSLS) problem (Garg, Balakrishnan et al., 2022). In the OSLS problem, the *detection*, *estimation* and *correction* problems can also be defined (Section 2.1) if we train an extra classifier to distinguish between the unseen class (zebra) and the seen classes (dog, cat, bird).

Relation with Class Imbalance Problem: It is worth noting that some special cases of the label shift problem also attract research attention. When all the categories in the test dataset from City B are evenly distributed, or are assumed to be evenly distributed, the Closed Set Label Shift problem degenerates to the *Class Imbalance problem* (Buda et al., 2018b; Lu et al., 2019; Wang, Liu et al., 2016). The objective of the Class Imbalance problem is to adjust the classifier trained on a class-imbalanced dataset to suit a class-uniform test dataset.

Relation with Long-Tailed Recognition: When the train set from City A has a highly imbalanced label distribution or even a Long-Tailed label distribution with some of the classes having only a few samples available, the Closed Set Label Shift problem becomes the *Long-Tailed Recognition* problem (Li, Cheung et al., 2022; Liu, Miao, Zhan, Wang, Gong and Yu, 2019). In the Long-Tailed Recognition task, the classifier is trained on a Long-Tailed dataset and tested on a class-uniform test set.

Relation with (Open) Domain Adaptation: The Closed Set Label Shift problem can be treated as a special case of the *Single Domain Generalization* problem in the domain adaptation task (Fan et al., 2021; Huang, Wang, Xing et al., 2020; Qiao et al., 2020; Wang, Luo et al., 2021). In the Single Domain Generalization problem, both the label frequency and the appearance of the animals are different between City A and City B. The *Single-Source Open-Domain Generalization* problem has a slightly different problem setup compared with the Open Set Label Shift problem, where City B is assumed to contain multiple extra OOD classes instead of a single OOD class (Bele et al., 2024).

1.2 A Brief History

Adjusting a machine learning model under distribution shift can be seen as an unsupervised domain adaptation task (Garg, Wu, Balakrishnan et al., 2020). Moreover, the domain adaptation setting can be further categorized as a transfer learning setting called transductive transfer learning (Pan and Yang, 2009).

Transfer learning or domain adaptation began receiving attention in the late 1990s with different names like “learning to learn" and “knowledge transfer" (Thrun and Pratt, 1998). The research field of domain adaptation aims to adapt a model trained on one domain (source) to perform well on another domain (target) where the data distributions are different. Early works primarily focused on situations where the entire distribution (both labels and features) shifts (Caruana, 1996).

In the 2000s, researchers started identifying specific forms of distribution shift, including covariate shift (where the distribution of data changes) (Shimodaira, 2000) and

Shift Name	Label Shift	Covariate Shift	Domain Shift
Shifted Distribution	$p(y)$ (label)	$p(x)$ (data)	$p(x,y)$ (joint)
Invariant Distribution	$p(x\|y)$	$p(y\|x)$	-

Table 1.1: Different types of distribution shift.

label shift (where the distribution of labels changes) (Saerens et al., 2002; Storkey, 2009; Vucetic and Obradovic, 2001).

By the early 2010s, label shift was formally recognized as a distinct type of distribution shift, which was referred to as the "Prior Probability Shift" (Chan and Ng, 2005; Storkey, 2009). During this time, several approaches are proposed for the label shift problem – the EM algorithm based approach (Saerens et al., 2002), the confusion matrix based approach (Vucetic and Obradovic, 2001) and the kernel mean matching approach (Zhang, Schölkopf et al., 2013).

The late 2010s saw further development of methods to address the label shift problem, with a focus on the more specific label shift *detection*, *estimation*, and *correction* problems (Lipton et al., 2018). Utilizing Deep Neural Networks as classifiers, researchers proposed various techniques to estimate and correct label shift in high dimensional datasets like CIFAR10/100. For instance, methods that leverage confusion matrices (Lipton et al., 2018) or classifier outputs (Alexandari et al., 2020) became popular.

In recent years, the label shift problem has been analyzed in broader and more realistic problem setups. For example, Wu, Guo, Su et al. (2021) analyze the label shift problem in an online learning setting, where the target label distribution can evolve with time; Garg, Balakrishnan et al. (2022) investigate the label shift problem in the open set classification setting, where the target domain has an extra out-of-distribution class. Details of the recent label shift literature will be reviewed in Chapter 2.

1.3 Terminology

This section briefly introduces the commonly used terminology in this thesis, which is related to the label shift problem and the three image classification tasks we consider.

Source/Target Domain: These terms are commonly used in the context of the label shift problem, or more broadly, the distribution shift problems. The source domain refers to the domain where the training data samples are collected, which are assumed to be drawn i.i.d. (independently and identically distributed) from the source data distribution. Similarly, the target domain refers to the domain where the test samples are collected, which are expected to be drawn i.i.d. from the target data distribution.

Closed Set Classification: A closed set classification task is a classification task where every sample in the test dataset has a corresponding ground truth label belonging to a class in the training dataset. In other words, the label set of the test data is either a subset of, or identical to, the label set of the training data.

Class Imbalance Problem: The class imbalance problem is commonly addressed in the closed set classification task context. When the training dataset from the source

domain has an imbalanced label distribution, the classifier tends to overfit the majority classes, which appear more frequently, while underfitting the minority classes, which are less represented (Buda et al., 2018b). **The class imbalance problem can be viewed as a special case of the label shift problem**, where the source domain has an imbalanced label distribution and the target domain has a uniform label distribution.

Long-Tailed Recognition: The Long-Tailed recognition task focuses on the closed set classification task, where the source domain training dataset has a Long-Tailed label distribution. In a Long-Tailed dataset, the majority (head) classes typically have thousands of samples, while the minority (tail) classes are represented by very few (Liu, Miao, Zhan, Wang, Gong and Yu, 2019). **The Long-Tailed recognition problem can be viewed as an extreme case of the class imbalance problem** and is frequently encountered in real-world applications (Xu et al., 2021).

Open Set Classification: The open set classification task extends the closed set classification task by introducing an extra out-of-distribution (OOD) class in the test data distribution (Chen, Peng et al., 2021). Consequently, the label set of the test data is the union of the label set of the training data (or a subset of it) and a label set with a single element representing the OOD class.

(Generalized) Zero-Shot Learning: In a zero-shot classification task, the challenge is to classify images where the labels in the test dataset belong to either "seen classes", which are present in the training dataset, or "unseen classes", which appear only in the test dataset (Elyor et al., 2017). This problem is typically tackled using semantic information that links the seen and unseen classes. Since the visual information for the unseen classes is unavailable during training, this type of image classification is referred to as Zero-Shot Learning. A generalized zero-shot learning (GZSL) task extends the traditional zero-shot learning (ZSL) task. The critical difference is that, in GZSL, the test set includes images from both seen and unseen classes (Xian, Lampert et al., 2019).

1.4 Objective and Approach

1.4.1 Objective

This thesis aims to *develop label shift estimation and correction models under different classification settings*. Considering existing works on the label shift problems, this thesis heads towards this objective via four steps:

1. Develop a robust label shift estimation model for label shift problems in the closed set classification tasks.
2. Improve the closed set label shift estimation model by relaxing the requirement of the model on the image classifier.
3. Analyze the label shift problem in the more realistic open set classification task and develop feasible models.
4. Explore the label shift problems in the more challenging Zero-Shot classification tasks.

Such an objective is beneficial because this series of investigations in increasingly more challenging or realistic settings can help the application of the label shift models to real world problems.

1.4.2 Scope

The scope of this thesis is restricted in several aspects. Firstly, we investigate exclusively the problem of label shift. Other types of distribution shift like covariate shift (Sugiyama et al., 2007), sub-population shift (Santurkar et al., 2020) or concept shift (Kull and Flach, 2014) are not considered or jointly considered with label shift in this thesis.

Moreover, the classification tasks we consider are limited to the single-label classification tasks, which require each data sample to be associated with one ground truth label that can be represented with a scalar. Other cases like multi-label classification tasks with multi-labels associated with one image (Tsoumakas and Katakis, 2008) and dense classification tasks like the semantic segmentation (Kirillov et al., 2023) and saliency detection (Zhang, Fan et al., 2021) task are not considered in this thesis.

Finally, the label shift problems we consider usually expect a pre-trained NN classifier or NN feature extractor to be available. Therefore, this thesis does not focus on developing methodology for NN training and optimization. The proposed models are compatible with pre-trained NN classifiers with different architectures.

1.4.3 Approach

The approach taken in this thesis is to first improve the robustness of the Closed Set Label Shift estimation model under highly imbalanced source and target label distributions by utilizing the Bayesian Inference methods (Chapter 3). A more general feature based Closed Set Label Shift estimation model is then developed by utilizing the Latent Variable Model (Chapter 4). The label shift problem in the open set classification task is explored through re-parameterizating the problem as a well-studied Closed Set Label Shift problem (Chapter 5). Finally, the label shift problem in the Zero-Shot Learning setting is tackled with a Neural Network trained with class-rebalancing loss and a Gaussian Process Regression model (Chapter 6).

1.5 Contribution Summary

The main contributions of this thesis are outlined below:

- A novel Closed Set Label Shift estimation model that is robust under a highly imbalanced source or target domain label distribution. The model consists of 1) an EM algorithm that obtains the Maximum *a Posteriori* (MAP) estimate of the target label distribution and 2) an Adaptive Prior Learning model that determines the parameter of the Bayesian prior used to obtain the MAP estimate.
- A novel, robust Closed Set Label Shift estimation model that is flexible with different choices of classifiers. The model consists of 1) an algorithm that obtains the MLE of the parameters of the class conditional feature distribution

and 2) two EM algorithms that obtain an MLE/MAP estimate of the target label distribution.

- A novel Open Set Label Shift estimation model that estimates the target label distribution with a pre-trained ID-class classifier and a pre-trained binary ID/OOD classifier without retraining or fine-tuning. The model consists of 1) an algorithm that obtains the estimate of the source domain ID data ratio, 2) an EM algorithm that obtains the MLE of target label distribution for ID classes and the OOD class and 3) an algorithm that obtains an estimate of the target ID data ratio with a relaxed assumption.
- A novel Zero-Shot learning model that is robust under source domain label shift for seen classes, which consists of 1) a feature re-balancing model trained with a novel class-balanced triplet loss and 2) a Gaussian Process Regression model that predicts unseen class feature prototypes based on the seen class features and the semantic information available.

1.6 Thesis Outline

This thesis comprises seven chapters, with Chapters 3 - 6 each including one of the four works (two published and two submitted) on the label shift problems for different image classification setups.

In **Chapter 2**, the formal definitions of the label shift problem in the closed set/open set classification tasks are introduced. Then, the methods used to tackle the label shift problems in this thesis – the Neural Networks models, Bayesian Inference method, EM algorithm, Gaussian Processes models and Latent Variable models, are briefly discussed. Finally, the literature on the label shift problem and related areas, including the Class Imbalance problem, Long-Tailed Recognition task, Out-of-distribution detection task and Zero-Shot learning task, are reviewed.

In **Chapter 3**, we investigate the label shift problem in the closed set classification task. A label shift estimation model named Maximum *a Posteriori* Label Shift (MAPLS) is proposed, which is robust when the source or target domain has a highly imbalanced label distribution. In the proposed model, an EM algorithm is derived and is proven to converge to the unique MAP estimate of the target label distribution under reasonable assumptions. An Adaptive Prior Learning (APL) model is proposed to determine the parameter of the Bayesian prior based on available data.

Chapter 4 also considers the Closed Set Label Shift estimation problem. A Generalized Label Shift Estimation (GLSE) framework is proposed to estimate the target label distribution based on a feature based classifier. In the proposed framework, four label shift estimation models are proposed based on the different types of image features being used. Two EM algorithms are derived and proved to converge to a MLE of the target label distribution and the unique MAP estimate of the target label distribution, which can be used in each of the four proposed models.

In **Chapter 5**, the Open Set Label Shift estimation problem is considered. An Open Set Label Shift for Maximum Likelihood Estimate (OSLS-MLE) model is proposed to estimate target label distribution for both the ID classes and the extra OOD class. The

proposed model includes 1) an estimate of the source label distribution of the OOD class, 2) an EM algorithm for Maximum Likelihood estimates (MLE) of the target label distribution, and 3) an estimate of the target label distribution of the OOD class under relaxed assumptions on the OOD classifier. The sampling errors of estimates in 1) and 3) are quantified with a concentration inequality.

In **Chapter 6**, the label shift problem in the Zero-Shot learning task is considered. A novel Zero-Shot Learning model is proposed, which is robust to label shift for the source domain seen classes. The proposed model consists of 1) a linear Neural Network that is trained with a label shift corrected loss that re-balances image features when the source domain has an imbalanced label distribution and 2) a Gaussian Process Regression model that predicts the image prototypes of the unseen classes based on the seen class image prototypes and the semantic vectors for the seen and unseen classes.

Finally, **Chapter 7** summarizes the main contributions of the thesis and discusses ongoing and future work stemming from this research.

1.7 Publications

This section provides the publications associated with my PhD research. Publications where I did not serve as the first author are not discussed in this thesis.

Published Works

- **Ye, Changkun**, Nick Barnes, Lars Petersson, and Russell Tsuchida. "Efficient Gaussian process model on class-imbalanced datasets for generalized zero-shot learning." In 2022 26th International Conference on Pattern Recognition (ICPR), pp. 2078-2085. IEEE, 2022.
- **Ye, Changkun**, Russell Tsuchida, Lars Petersson, and Nick Barnes. "Label shift estimation for class-imbalance problem: A bayesian approach." In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 1073-1082. 2024.
- **Ye, Changkun**, Russell Tsuchida, Lars Petersson, and Nick Barnes. "Open Set Label Shift with Test Time Out-of-Distribution Reference." In Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR), 2025, pp. 30619-30629.

Under Review Papers

- **Ye, Changkun**, Russell Tsuchida, Lars Petersson, and Nick Barnes. "Label Shift for Class Imbalance Problem – A General Two Stage Framework." submitted to TPAMI.

Non-lead Author Publications

- Liu, Jiawei, **Changkun Ye**, Ruikai Cui, and Nick Barnes. "Self-Calibrating Vicinal Risk Minimisation for Model Calibration." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3335-3345. 2024.

- Han, Pengxiao, **Changkun Ye**, Jieming Zhou, Jing Zhang, Jie Hong, and Xuesong Li. "Latent-based Diffusion Model for Long-tailed Recognition." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2639-2648. 2024.
- Liu, Jiawei, **Changkun Ye**, Shan Wang, Ruikai Cui, Jing Zhang, Kaihao Zhang, and Nick Barnes. "Model calibration in dense classification with adaptive label perturbation." In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 1173-1184. 2023.

Chapter 2

Background and Literature Review

This chapter introduces the formal definition of the label shift problem, reviews the background material to clarify key terminology, and outlines the related research literature. Formal definitions of the label shift problems in the closed set and the open set classification tasks are first provided in Section 2.1, as they are the main problems investigated in this thesis.

The background section (Section 2.2) introduces the Deep Learning and Machine Learning methods used in this thesis. These methods include 1) the Neural Network models that are used as image classifiers or image feature extractors, 2) the Bayesian inference methods, the EM algorithm and the Latent Variable Model (LVM) used in the closed set and Open Set Label Shift estimation models proposed in Chapter 3,4,5 and 3) the Gaussian Process models that are used in the label shift correction problem in the ZSL task (Chapter 6).

The literature review section (Section 2.3) briefly summarises recent literature on 1) the label shift problem in different classification tasks, 2) the class imbalance problem and the Long-Tailed recognition problem, which can be seen as a special case of the label shift problem, 3) the OOD detection task, where the OOD models proposed in these works are used in our Open Set Label Shift models (Chapter 5) and 4) the ZSL task that could also suffer from the problem of label shift (Chapter 6).

2.1 Problem Definitions

This section provides the formal definitions of the closed set and the open set label shift problems. The Closed Set Label Shift *detection*, *estimation* and *correction* problem are defined following the definition in Lipton et al. (2018). The Open Set Label Shift problem is defined by extending the closed set classification setting to the open set classification setting. The Closed Set Label Shift *estimation* and *correction* problem are analyzed in Chapter 3,4. The Open Set Label Shift *estimation* and *correction* problem are investigated in Chapter 5.

2.1.1 Closed Set Label Shift Problem

The Closed Set Label Shift (CSLS) problem considers the closed set classification task where the source (train) and target (test) domain have different label distributions $p_s(y) \neq p_t(y)$. The objective of the CSLS problem is to *detect* if label shift happens

between the source and target domain, *estimate* target label distribution and *correct* a source domain classifier to the target domain under label shift.

Assumption: To analyze the CSLS problem, it is usually assumed the conditional distribution of data given label $p(x|y)$ is invariant between the source and target domain (Alexandari et al., 2020; Garg, Wu, Balakrishnan et al., 2020; Lipton et al., 2018), which can be formally stated as

Assumption 1. (Closed Set Label Shift Assumption)

$$p_s(x|y=i) = p_t(x|y=i) \quad \text{for all} \quad i \in \mathcal{Y}. \tag{2.1}$$

Datasets: In the CSLS problem, we have two observed datasets: a labeled source domain (train) dataset $\mathcal{D}^s$ and an unlabeled target domain (test) dataset $\mathcal{D}^t$. The two datasets can be formally defined as:

$$\begin{cases} \mathcal{D}^s = \{x_i^s, y_i^s\}_{i=1}^{N^s} & \text{where} \quad (x_i^s, y_i^s) \sim_{i.i.d.} p_s(x,y) \\ \mathcal{D}^s = \{x_i^t\}_{i=1}^{N^t} & \text{where} \quad (x_i^t, \cdot) \sim_{i.i.d.} p_t(x,y). \end{cases} \tag{2.2}$$

Model: As a classification problem, the CSLS problem usually assumes a source domain classifier or feature extractor f is available.

Problem Setup: Under Assumption 1, given source labeled dataset $\mathcal{D}^s$, target unlabeled dataset $\mathcal{D}^t$ and classifier f, the CSLS problem focus mainly on three sub-problems, named label shift *detection*, *estimation* and *correction* – 1) ***detection***: verify if the label distributions in the source and target domain are identical, *i.e.* $p_s(y) = p_t(y)$, 2) ***estimation***: estimate the target label distribution $p_t(y=\cdot)$ and 3) ***correction***: adapt a source domain classifier f to the target domain. The formal definition of the CSLS problem is as follows:

Definition 1. (**Closed Set Label Shift Problem**)

Under Assumption 1, given:

- Source domain labeled data $\mathcal{D}^s = \{(x_i^s, y_i^s)\}_{i=1}^{N^s}$ (K classes);
- Target domain unlabeled data $\mathcal{D}^t = \{x_i^t\}_{i=1}^{N^t}$ (K classes);
- Source domain classifier f.

the CSLS problem is to solve

- *Detection*: Verify $p_s(y=\cdot) = p_t(y=\cdot)$;
- *Estimation:* Estimate $p_t(y=\cdot)$;
- *Correction:* Model $p_t(y=\cdot|x)$ based on f.

Assumption 1 and Definition 1 will be referred to in the later chapters when the Closed Set Label Shift problem is under consideration.

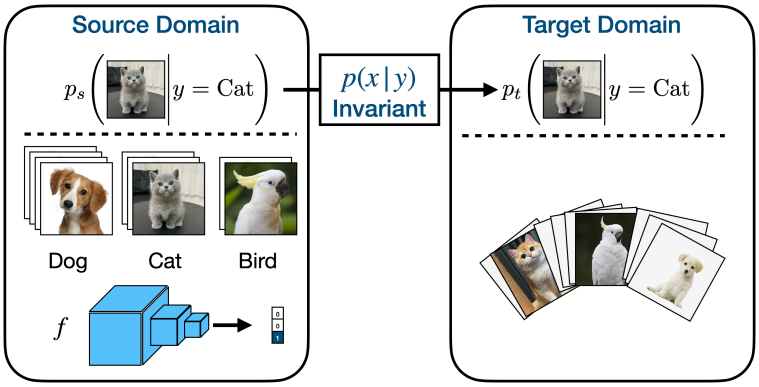

Figure 2.1: Available information in the Closed Set Label Shift problems, which includes: 1) **assumption**: conditional distribution of image given label is the same between the source and target domain (Assumption 1), 2) **datasets**: a source domain labeled dataset and a target domain unlabeled dataset and 3) **model**: a source domain classifier f. This information is used for closed label shift ***detection***, ***estimation*** and ***correction*** problems.

2.1.2 Open Set Label Shift Problem

The Open Set Label Shift (OSLS) problem has only been studied recently (Garg, Balakrishnan et al., 2022). The OSLS problem extends the closed set setting, where the target domain has an extra out-of-distribution (OOD) class. In the OSLS problem, we have the data space $\mathcal{X} \subseteq \mathbb{R}^d$ and label space $\mathcal{Y} \cup \{K+1\}$ that includes in-distribution (ID) class labels $\mathcal{Y} = \{1, 2, .., K\}$ and the OOD class label $K+1$.

Assumption: Similar to the Closed Set Label Shift assumption (Assumption 1), the Open Set Label Shift assumption can be formally stated as

Assumption 2. (Open Set Label Shift Assumption)

$$p_s(x|y=i) = p_t(x|y=i) \quad \text{for all} \quad i \in \mathcal{Y} \cup \{K+1\}, \tag{2.3}$$

where the OOD data in the source domain should also look the same as the OOD data in the target domain, even when no source domain OOD samples are available in the OSLS problem setup.

Data: Similar to the CSLS problem, the OSLS problem also has two observed datasets: a source domain dataset $\mathcal{D}^s$ and a target domain dataset $\mathcal{D}^t$. The only difference is that the target domain dataset $\mathcal{D}^t$ will contain samples that belong to K ID classes and 1 OOD class. The two datasets can be formally defined as:

$$\begin{cases} \mathcal{D}^s = \{x_i^s, y_i^s\}_{i=1}^{N^s} & \text{where} \quad (x_i^s, y_i^s) \sim_{i.i.d.} p_s(x,y) \\ \mathcal{D}^t = \{x_i^t\}_{i=1}^{N^t} & \text{where} \quad (x_i^t, \cdot) \sim_{i.i.d.} p_t(x,y), \end{cases} \tag{2.4}$$

where $p_t(x,y)$ is supported on $\mathcal{X} \times \mathcal{Y} \cup \{K+1\}$ with the extra OOD class.

Model: the OSLS problem assumes the availability of a source domain ID classifier $f : \mathcal{X} \to \Delta^{K-1}$ and an OOD/ID classifier $h : \mathcal{X} \to [0,1]$.

Problem Setup The OSLS problem can also be divided into *detection*, *estimation* and *correction* problems – 1) ***detection***: verify if the source domain label distribution is identical to the target domain label distribution for ID classes, *i.e.* $p_s(y = \cdot) = p_t(y = \cdot | y \in \mathcal{Y})$, 2) ***estimation***: estimate the target label distribution $p_t(y = \cdot)$ for $K + 1$ classes and 3) ***correction***: construct a model for the target domain $p_t(y|x)$ based on the source domain ID classifier f and the source domain ID/OOD classifier h. The formal definition of the OSLS problem can be stated as follows:

Definition 2. (**Open Set Label Shift Problem**)

Under Assumption 2, given:

- Source domain ID labeled data $\mathcal{D}^s = \{(x_i^s, y_i^s)\}_{i=1}^{N^s}$ (K ID classes);
- Target domain unlabeled data $\mathcal{D}^t = \{x_i^t\}_{i=1}^{N^t}$ (K ID classes + 1 OOD class);
- Source domain ID classifier f;
- Source domain ID/OOD classifier h,

the Open Set Label Shift problem is to solve

- *Detection*: Verify $p_s(y) = p_t(y|y \in \mathcal{Y})$;
- *Estimation:* Estimate $p_t(y = \cdot)$;
- *Correction:* Model $p_t(y = \cdot|x)$ based on f and h.

Assumption 2 and Definition 2 will be referred to in the later chapters when the Open Set Label Shift problem is under consideration.

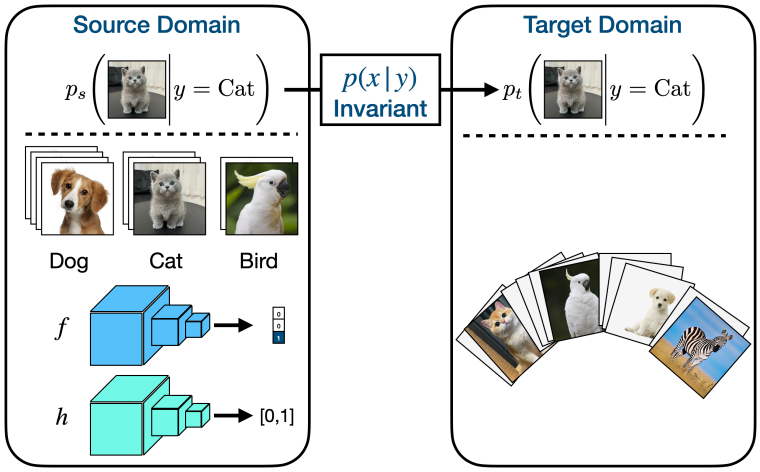

Figure 2.2: Available information in the Open Set Label Shift problems, which includes: 1) **assumption**: conditional distribution of image given label is the same between the source and target domain (Assumption 2), 2) **datasets**: a source domain labeled dataset and a target domain unlabeled dataset (that includes an extra OOD class) and 3) **models**: a source domain classifier f for ID classes and a source domain ID/OOD binary classifier h. This information is used for Open Set Label Shift ***detection***, ***estimation*** and ***correction*** problems.

2.2 Background Materials

This section outlines the literature and techniques that are used in our work for the label shift problem. The Neural Network models are used as image classifiers or feature extractors. Several Statistical Inference methods, Bayesian Inference methods, EM algorithm and Latent Variable Models are used in the closed set and Open Set Label Shift estimation and correction models in Chapter 3,4,5. The Gaussian Process model is used in the Zero-Shot Learning task under label shift, which is discussed in Chapter 6.

Method	Used in	Functionalities
Neural Networks	Chapter 3,4,5,6	image classifier/feature extractor
Statistical Inference	Chapter 3,4,5	parameter estimation
Bayesian Inference	Chapter 3,4,5,6	parameter estimation
Latent Variable Model	Chapter 4	formulating the problem
EM algorithm	Chapter 3,4,5	optimizing the (non) Bayesian objective
Gaussian Process (Regression)	Chapter 6	predict class prototypes

Table 2.1: Summary of the Machine/Deep Learning methods used in this thesis.

2.2.1 Neural Network Models

This section briefly introduces the Neural Network (NN) models, which are used as image classifiers or feature extractors in the label shift models proposed in the following chapters (Chapter 3, 4, 5). Please refer to Goodfellow (2016) for a more detailed discussion about the NN models and deep learning methods.

Neural Network (NN) models have performed excellently in many machine learning problems in recent years (He, Zhang et al., 2016; Russakovsky et al., 2015a; Simonyan and Zisserman, 2015). In a multi-layer feedforward neural network, the input data will pass through a series of blocks, each consisting of a linear transformation function and a non-linear activation function. For example, a L-layer fully-connected neural network model $f(x)$ can be defined as:

$$f(x) = f^{(L)}(x), \text{ where } f^{(l)}(x) = \begin{cases} \sigma(\boldsymbol{W}^{(l)} f^{(l-1)}(x) + \boldsymbol{b}^{(l)}), & l \in \{2, 3, ..., L\}, \\ \sigma(\boldsymbol{W}^{(1)} x + \boldsymbol{b}^{(l)}), & l = 1, \end{cases} \tag{2.5}$$

with $\boldsymbol{W}^{(l)}, \boldsymbol{b}^{(l)}$ as the weight matrix and bias vector in the l^{th} layer of the network and $\sigma(\cdot)$ as the non-linear activation function.

The NN models are usually trained on a dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$ drawn i.i.d. from some data distribution $p(x, y)$ with a loss function $\ell(\cdot, \cdot)$. The objective is to minimize the expected risk $R(f)$ w.r.t. model f defined in Eq. (2.5), which is approximated by the empirical risk (Vapnik, 1991; Zhang, Cisse et al., 2018):

$$R(f) = \int_{\mathcal{X} \times \mathcal{Y}} p(x, y) \cdot \ell(f(x), y) dx dy \approx \frac{1}{N} \sum_{i=1}^{N} \ell(f(x_i), y_i). \tag{2.6}$$

The optimization of the empirical risk is achieved via the stochastic gradient descent (SGD) method, where in each iteration, the NN model f is updated by backpropagating the gradient of the loss function $\ell(\cdot,\cdot)$ on a random subset of $\mathcal{D}$ over model parameters $\{\boldsymbol{W}^{(l)}, \boldsymbol{b}^{(l)}\}_{l=1}^{L}$. With SGD, the NN model f can converge to local optima, which, in many cases, can be sufficient for the problem.

Three types of NN models are mainly considered in this work are listed below:

SoftMax Classifier $f : \mathcal{X} \to \Delta^{K-1}$ that usually takes the form:

$$f(x)_i = \frac{e^{f^{(L)}(x)_i}}{\sum_{l=1}^{K} e^{f^{(L)}(x)_l}} \quad \text{for all} \quad i \in \{1, 2, ..., K\}. \tag{2.7}$$

Discrete Classifier $f : \mathcal{X} \to \{1, 2, ..., K\}$ that usually takes the form:

$$f(x) = \underset{i \in \{1,2,...,K\}}{\arg\max} \frac{e^{f^{(L)}(x)_i}}{\sum_{l=1}^{K} e^{f^{(L)}(x)_l}}. \tag{2.8}$$

Feature Extractor $f : \mathcal{X} \to \mathcal{Z} \subseteq \mathbb{R}^L$ that usually takes the form:

$$f(x) = f^{(L-1)}(x), \tag{2.9}$$

where $f^{(L-1)}(x)$ is the penultimate layer output of a NN classifier.

2.2.2 Statistical Inference

Statistical inference usually aims to infer the properties of an underlying probability distribution given observed data from that distribution (Upton and Cook, 2014). This section provides a brief introduction about a specific statistical inference method – the Maximum Likelihood Estimation method, as this method will be used in the label shift estimation models in Chapter 4, 5. A short introduction about the concentration inequalities is also provided, which are used to quantify the probabilistic error bound of the statistical estimators in Chapter 5. Please see Lehmann and Casella (2006) for a more detailed discussion of the statistical inference methods.

Maximum Likelihood Estimation: Maximum Likelihood estimation is a method used to estimate the parameters of an assumed probability distribution based on the observed data. Given the observed dataset $\mathcal{D} = \{x_i\}_{i=1}^{N}$, if we assume the data samples in $\mathcal{D}$ are drawn i.i.d. from some probability distribution $p(x)$ that falls within a parametric family $\mathcal{Q} = \{f(\cdot;\theta)|\theta \in \Theta\}$, *i.e.* $p(x) \in \mathcal{Q}$, the likelihood function of the parameter θ given the observed data $\mathcal{D}$ can be written as:

$$L(\theta; \mathcal{D}) = \prod_{i=1}^{N} f(x_i; \theta). \tag{2.10}$$

The Maximum Likelihood estimate (MLE) of θ given the observed data $\mathcal{D}$ can be obtained by maximizing the likelihood w.r.t. $\theta \in \Theta$, or equivalently, minimizing the negative log likelihood w.r.t. $\theta \in \Theta$:

$$\theta^{\text{MLE}} = \arg\min_{\theta \in \Theta} -\log L(\theta; \mathcal{D}) = \arg\min_{\theta \in \Theta} -\sum_{i=1}^{N} \log f(x_i; \theta). \tag{2.11}$$

The objective in Eq. (2.11) is a constraint optimization problem and thus can be solved with related optimization methods. For example, when f is differentiable and the constraint $\theta \in \Theta$ can be rewritten as an equality constraint, a local optimal of the objective can be obtained with the Lagrange multiplier method.

One nice property of the Maximum Likelihood Estimation method is that, MLE is invariant under reparameterization (Murphy, 2012). Or more specifically, if $L(\theta; \mathcal{D})$ is our likelihood function of the parameter θ and $L(\eta; \mathcal{D})$ is the reparametrization of $L(\theta; \mathcal{D})$ with parameter $\eta = g(\theta)$, then the MLE of η can be obtained with the MLE of θ under reparameterization:

$$\eta^{\text{MLE}} = g(\theta^{\text{MLE}}). \tag{2.12}$$

Concentration Inequalities: The concentration inequalities provide upper or lower bounds on the probability of a random variable deviating from a certain value. This thesis uses a common concentration inequality – Hoeffding's inequality. For a series of independent random variables $X_1, X_2, ..., X_n$ that satisfy 1) $a_i \leq X_i \leq b_i$ and 2) $b_i - a_i < C$ for all $i = 1, 2, ..., n$, Hoeffding's inequality says that for a small $t > 0$, with high probability of at least $1 - 2\exp\left(-n \cdot t^2 / C^2\right)$ we have:

$$\left| \frac{1}{n} \sum_{i=1}^{n} X_i - \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}[X_i] \right| < t. \tag{2.13}$$

Hoeffding's inequality says that for the samples drawn i.i.d. from a distribution supported on a bounded interval, there is a large probability that the sample mean deviates from the true mean by a small amount, where such deviation decreases exponentially in the sample number n and the probability increases in n.

The concentration inequality can be used to quantify the error bound of a statistical estimator, *e.g.* Maximum Likelihood Estimator.

2.2.3 Bayesian Inference

This section provides a brief introduction about the Bayesian inference methods used in this thesis, including the Maximum *a Posteriori* (MAP) Estimator and the Markov Chain Monte Carlo method. Other methods like conjugate priors and variational inference are also introduced for completeness. A more detailed discussion about the Bayesian inference methods can be found in Box and Tiao (2011).

Bayesian Inference is a well-known statistical inference method that provides a theoretical framework to combine prior information with observed data (Bishop and Nasrabadi, 2006; Box and Tiao, 2011). We consider the probability distribution of a parameter $\theta \in \Theta$ conditioned on observed data $\mathcal{D}$, given 1) prior information about θ before any data is observed and 2) the so-called likelihood function of the observed data given θ, according to the Bayes' Theorem:

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})}, \tag{2.14}$$

where:

- $p(\theta)$ is the prior probability distribution of the parameter over the parameter space $\theta \in \Theta$. $p(\theta)$ encodes information that we know about θ before any data is observed.
- $p(\mathcal{D}|\theta)$ is the probability distribution of observing data $\mathcal{D}$ conditioned on the parameter θ. For a fixed θ, $p(\mathcal{D}|\theta)$ is a function of the observed data $\mathcal{D}$.
- $p(\mathcal{D})$, usually referred as "evidence", is the probability that the observed data $\mathcal{D}$ is drawn i.i.d. from the marginal distribution of data. Although $p(\mathcal{D})$ can be computed via the integral

  $$p(\mathcal{D}) = \int_{\Theta} p(\mathcal{D}|\theta)p(\theta)d\theta, \tag{2.15}$$

  this integral is usually computationally expensive or even intractable in practice.
- $p(\theta|\mathcal{D})$ is what we want – the posterior distribution of the parameter given observed data.

In some simple cases (*e.g.* conjugate prior), the analytical expression of the posterior $p(\theta|\mathcal{D})$ can be computed easily. However, this is not the case for most practical problems because the integral Eq. (2.15) in the evidence term is usually computationally expensive or even intractable. This work outlines several methods commonly used in Bayesian Inference:

- **Conjugate Priors**: for certain types of likelihood function, the posterior distribution $p(\theta|\mathcal{D})$ is in the same family of the prior distribution $p(\theta)$ and thus can be obtained with a closed form solution;
- **Point Estimators**: the Bayesian point estimators computes the point estimates of θ instead of obtaining the original posteriors. The point estimates can be obtained without knowing $p(\mathcal{D})$;
- **Markov Chain Monte Carlo (MCMC)**: MCMC methods can be used in Bayesian Inference to draw i.i.d. samples from the posterior without knowing the evidence;

We outline the technical details of the above methods in the following few pages.

Conjugate Priors:

Given a likelihood $p(\mathcal{D} \mid \theta)$ and a prior $p(\theta)$ belonging to some class Π, the class Π is said to be conjugate for the likelihood if $p(\theta \mid \mathcal{D})$ also belongs to Π. In many cases, the parameters of the posteriors can be updated in order to update the posterior. The problem of Bayesian Inference is then finding the parameter of the posterior distribution in the conjugate class. We summarize several commonly used Conjugate Prior classes below:

Likelihood $p(\mathcal{D}\|\theta)$	Prior $p(\theta)$	Posterior $p(\theta\|\mathcal{D})$
Categorical Cat$(K, \boldsymbol{\pi})$	Dirichlet Dir$(K, \boldsymbol{\alpha})$	Dirichlet Dir$(K, \boldsymbol{\alpha}')$
Gaussian $\mathcal{N}(\mu_1, \Sigma_1)$	Gaussian $\mathcal{N}(\mu_2, \Sigma_2)$	Gaussian $\mathcal{N}(\mu', \Sigma')$

Table 2.2: Conjugate Prior examples that are commonly used.

For a complete summary of the Conjugate Priors, please refer to the literature in Bayesian Statistics (Box and Tiao, 2011; Murphy, 2012).

Point Estimators:

In Bayesian Inference, given the posterior distribution $p(\theta|\mathcal{D})$, several point estimators are available:

- **Posterior mean** is the point estimate that minimizes an expected mean square loss over the posterior distribution, the optimization objective can be written as:

$$\theta^* = \arg\min_{\theta \in \Theta} \int_{\theta_0 \in \Theta} p(\theta_0|\mathcal{D}) \cdot \|\theta_0 - \theta\|_2^2 d\theta_0. \tag{2.16}$$

- **Posterior median** is the point estimate that minimizes an expected L1 loss over the posterior distribution, the optimization objective can be written as:

$$\theta^* = \arg\min_{\theta \in \Theta} \int_{\theta_0 \in \Theta} p(\theta_0|\mathcal{D}) \cdot \|\theta_0 - \theta\|_1 d\theta_0. \tag{2.17}$$

- **Maximum** *a Posteriori* **(MAP)** estimate simply finds a θ^* that maximizes the posterior:

$$\theta^* \in \arg\max_{\theta \in \Theta} p(\theta|\mathcal{D}). \tag{2.18}$$

  Under mild conditions, the Bernstein - von Mises theorem ensures that the MAP estimate converges asymptotically in the number of data samples to the MLE (2.11);

MCMC:

Markov Chain Monte Carlo (MCMC) method is a class of algorithms that draw samples from a probability distribution (Brooks, 1998; Geyer, 1992). Given an analytical probability distribution, MCMC constructs a Markov chain such that the equilibrium distribution of the chain equals the objective distribution. When sufficient samples are obtained from the Markov chain, the samples at the end of the chain can be seen as i.i.d. samples drawn from the objective distribution.

In Bayesian Inference, MCMC is usually used when the posterior is complicated or has a high dimensional parameter space Θ. In these cases, the moments of the posterior might provide sufficient information that researchers are interested in, and MCMC can help collect i.i.d. samples from the posterior.

Early models of MCMC include naive Metropolis–Hastings and Gibbs sampling. More advanced MCMC algorithms have been proposed recently, including the Metropolis-adjusted Langevin algorithm (Dwivedi et al., 2018), Hamiltonian Monte-Carlo (Betancourt, 2017; Neal, 2011) and Non-U-Turn Sampler (Hoffman, Gelman et al., 2014), which claims better sampling efficiency or easier to use setup.

Method	Output	Pros	Cons
Conjugate Prior	exact posterior	exact	restrictions on likelihood and prior
Point Estimate	point	fast	discard most posterior information
MCMC	posterior i.i.d. samples	asymptotically exact	less efficient (memory, computation)

Table 2.3: Summary of the outputs, pros and cons of the Bayesian Inference Methods used in this thesis.

2.2.4 Latent Variable Model

Latent Variable Models (LVMs) are proposed to model complicated relations of observed variables with simple relations between this variable and some unobserved variables. Depending on whether the variables are discrete (also known as categorical) or continuous (also known as metrical), the LVMs are usually divided into four categories, as shown in Tab. 2.4.

Latent \ Observed	metrical	categorical
metrical	Factor analysis	Latent trait analysis
categorical	Latent profile analysis	Latent class analysis

Table 2.4: Summary of different types of Latent Variable Models.

In the LVMs, the observed variables $X_1, X_2, ..., X_p$ are assumed to be independently conditioned on the unobserved variable $Y_1, Y_2, ..., Y_q$, where $q \ll p$ to ensure the model is valid. Please refer to https://www.stats.ox.ac.uk/~steffen/teaching/fsmHT07/fsm607c.pdf for detailed discussion. This thesis mainly utilizes the Latent Profile Analysis (LPA) model, where the observed variables $X_1, X_2, ..., X_p$ are supported on a continuous space $\mathcal{X}$ and the latent variables $Y_1, Y_2, ..., Y_q$ are supported on a discrete space $\mathcal{Y}$.

Mixture Model and EM algorithm: When the unobserved latent variable is categorical (*e.g.* $Y \sim \text{Cat}(K, \boldsymbol{\pi})$), and the conditional variable $X|Y = i$ for each category i belongs to the same distribution family (*e.g.* $X|Y = i \sim \mathcal{N}(\mu, \Sigma)$ for all $i \in \{1, 2, ..., K\}$), then the Latent Profile Analysis model degenerates to the Mixture

Model. In Mixture Models, EM algorithm is a famous method to estimate the parameter of the mixture weight, *i.e.* parameter of the marginal distribution of the latent variable $p(y = \cdot) = \boldsymbol{\pi}$. A more detailed discussion about a particular case of the Mixture Model – the Gaussian Mixture Model, can be found in Chapter 9 in Bishop and Nasrabadi (2006) .

2.2.5 EM Algorithm

EM algorithm is a method widely used in the Latent Variable Models. This section introduces the EM algorithm and discusses its advantages. In the following chapters (Chapter 3, 4, 5), different EM algorithms are derived in the proposed label shift estimation models.

The Expectation–Maximization (EM) algorithm is an iterative method to find MLE, or MAP estimates θ^* of parameters θ (Moon, 1996). That is

$$\theta^* = \arg\max_{\theta} L(\theta; X),$$

where $L(\theta; X)$ is a log posterior of θ given X or a log-likelihood of X given θ. The algorithm works when the statistical model has unobserved latent variables Y.

An EM algorithm consists of two steps in every iteration, namely the **Expectation-Step** and **Maximization-Step**, which are usually referred to as the **E-Step** and **M-Step**, respectively. To derive an EM algorithm that maximises $L(\theta; X)$, we first write the analytical expression of the log posterior or log likelihood of parameter θ given observed variable X and unobserved latent variable Z as $L(\theta; X, Y)$.

In the **E-Step**, the model constructs a $Q(\theta|\theta^{(t)})$ as the expectation of $L(\theta; X, Y)$ w.r.t. latent variable Y given observed variable X and current $\theta^{(t)}$:

$$Q(\theta|\theta^{(t)}) = \mathop{\mathbb{E}}_{Y|X,\theta^{(t)}}[L(\theta; X, Y)]. \tag{2.19}$$

In the **M-Step**, the model finds the optimal $\theta^{(t+1)}$ with:

$$\theta^{(t+1)} = \arg\max_{\theta} Q(\theta|\theta^{(t)}). \tag{2.20}$$

By repeating the two steps until convergence, under mild conditions, the algorithm is guaranteed to converge to a stationary point of $L(\theta; X)$ under mild conditions. We recommend the following lecture note: `http://www.columbia.edu/~mh2078/MachineLearningORFE/EM_Algorithm.pdf` for more detailed proof and discussion.

Some advantages of the EM algorithm are:

- The latent variable may be difficult to integrate out. The EM algorithm allows one to manipulate conditional distributions by marginalising out a certain log likelihood to the probability measure of a latent variable conditioned on data and parameters, which is often easier than integrating the latent variable directly.

- The **M-Step** often admits a closed form solution.
- It is guaranteed that $L(\theta; X) - L(\theta^{(t)}; X) \geq Q(\theta|\theta^{(t)}) - Q(\theta^{(t)}|\theta^{(t)})$, which means improvement in $L(\theta; X)$ will not be less than improvement in $Q(\theta|\theta^{(t)})$ by solve for best $\theta^{(t+1)}$ in the t^{th} **M-Step**.

2.2.6 Gaussian Process

Gaussian Process (GP) is a statistical process in which any finite collection of random variables follows a multi-variate Gaussian distribution (Williams and Rasmussen, 2006), which can be seen as a distribution of functions over a continuous space. GP models have been used for both regression and classification problems (*e.g.* Hensman et al. (2015); Urtasun and Darrell (2007)), even on the large scale datasets (Lawrence, 2004). This dissertation uses the GP regression model in the Open Set Label Shift problem.

Formally speaking, a Gaussian process can be defined by a mean function and a covariance function, which we denote as:

$$\begin{aligned} m(x) &= \mathbb{E}[f(x)] \\ k(x, x') &= \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))], \end{aligned} \tag{2.21}$$

and then the GP can be written as:

$$f(x) = \mathcal{GP}(\mu(x), k(x, x')), \tag{2.22}$$

where $m(\cdot)$ is a mean function and $k(\cdot, \cdot)$ is a kernel function.

Given a training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, the GP regression model defined in Eq. (2.22) says that the ground truth output data $\boldsymbol{y} = [y_1, y_2, ..., y_N]^T$ in $\mathcal{D}$ follows a multi-variate Gaussian distribution

$$\boldsymbol{y} \sim \mathcal{N}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}|\theta) + \sigma^2 \mathbf{I}) \tag{2.23}$$

where $\boldsymbol{x} = [x_1, x_2, ..., x_N]^T$ denotes the ground truth input data in $\mathcal{D}$.

In the test time, given the test sample x^*, the GP regression model says the posterior of the output $y^* = f(x^*)$ given the input x^* and training dataset $\mathcal{D}$ also follows a multi-variate Gaussian distribution $f(x)|x^*, \mathcal{D} \sim \mathcal{N}(\mu^*, \sigma^*)$, with:

$$\begin{aligned} \mu^* &= k(x^*, \boldsymbol{x})^T \left[k(\boldsymbol{x}, \boldsymbol{x}) + \sigma^2 \mathbf{I}\right]^{-1} \boldsymbol{y} \\ \sigma^* &= k(x^*, \boldsymbol{x})^T \left[k(\boldsymbol{x}, \boldsymbol{x}) + \sigma^2 \mathbf{I}\right]^{-1} k(\boldsymbol{x}, x^*), \end{aligned} \tag{2.24}$$

where $\boldsymbol{x} = [x_1, x_2, ..., x_N]^T, \boldsymbol{y} = [y_1, y_2, ..., y_N]^T$, $\mathbf{I}$ is the N-dimensional identity matrix and

$$\begin{aligned} k(x^*, \boldsymbol{x}) &= [k(x^*, x_1), k(x^*, x_2), ..., k(x^*, x_N)]^T, \\ k(\boldsymbol{x}, x^*) &= [k(x_1, x^*), k(x_2, x^*), ..., k(x_N, x^*)]^T, \\ k(\boldsymbol{x}, \boldsymbol{x}) &= \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_N) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_N) \\ \vdots & \vdots & \vdots & \vdots \\ k(x_N, x_1) & k(x_N, x_2) & \dots & k(x_N, x_N) \end{bmatrix}. \end{aligned} \tag{2.25}$$

In practice, the mean function $m(\cdot)$ is usually set as $m(\cdot) \equiv 0$, and different positive semi-definite kernel functions $k(\cdot, \cdot)$ are used. When kernel function $k(\cdot, \cdot|\theta)$ has hyper-parameter $\theta \in \Theta$, θ can be trained by maximizing the log marginal likelihood of the data $\boldsymbol{x}, \boldsymbol{y}$ in $\mathcal{D}$, which satisfies $\boldsymbol{y} \sim \mathcal{N}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}|\theta) + \sigma^2\mathbf{I})$:

$$\theta = \arg\max_{\theta \in \Theta} \left(-\frac{1}{2} \boldsymbol{y}^T \left[k(\boldsymbol{x}, \boldsymbol{x}|\theta) + \sigma^2 \mathbf{I} \right]^{-1} \boldsymbol{y} - \frac{1}{2} \log \det(k(\boldsymbol{x}, \boldsymbol{x}|\theta)) \right), \tag{2.26}$$

where $\boldsymbol{x}, \boldsymbol{y}$ contains all data points in $\mathcal{D}$ and $\det(\cdot)$ returns the determinant of a matrix.

The GP regression model (Eq. (2.24)) is a linear model, and the predicted μ^* is given by a linear combination of $\boldsymbol{y}$ in the dataset $\mathcal{D}$. GP works well for small-scale datasets (N not large) and can provide prediction with uncertainty. However, for large-scale datasets, GP can suffer from high computational cost due to the $N \times N$ matrix inversion in Eq. (2.24).

2.3 Literature Review

This section briefly introduces existing works in the label shift problem and several related classification problems. The literature review on the label shift problem mainly consists of publications on the label shift *detection*, *estimation* and *correction* problems in the closed set classification tasks (Section 2.3.1). Recent works on the other more challenging label shift settings, like active learning under label shift, are also summarized. The literature on the class imbalance problem and the Long-Tailed Recognition problem are reviewed in Section 2.3.2,2.3.3 respectively, as these problems can be seen as a special or an extreme case of the label shift problem. Finally, the literature on the open world classification tasks like Out-of-Distribution detection (Section 2.3.4) and Zero-Shot Classification (Section 2.3.5) are introduced because this thesis will also consider the label shift problem in the open world classification settings. The chapters that are related to the literature of each problem are summarized in Tab. 2.5.

Method	Related to	Usage
Label Shift Problem	Chapter 3,4,5,6	Baselines
Class Imbalance Problem	Chapter 3,4,5,6	Experimental Setup
Long-Tailed Recognition	Chapter 3,4,5	Experimental Setup
Out-of-Distribution Detection	Chapter 5	Backbone Classifier
Zero-Shot Classification	Chapter 6	Problem Setup

Table 2.5: Summary of the related literature in this thesis.

2.3.1 Label Shift Problem

We provide a brief introduction of the recent literature on the Closed Set Label Shift *detection*, *estimation* and *correction* problem and some more challenging label shift settings (*e.g.* open set). A summary of the these works and the methods proposed in this thesis are provided in Tab 2.6.

2.3.1.1 Closed Set Label Shift Detection

Few works have focused on the CSLS detection task. The most recent one is BBSD proposed by Lipton et al. (2018). The general idea of the model is — when no label shift happens; the classifier should behave similarly over the source and target domain datasets under the label shift assumption (Assumption 1).

2.3.1.2 Closed Set Label Shift Estimation

BBSE: Lipton et al. (2018) proposed the BBSE model for CSLS estimation with a discrete classifier $f : \mathcal{X} \to \mathcal{Y} = \{1, 2, ..., K\}$. The general idea of the model is to utilize the CSLS Assumption 1 to construct a series of linear equations with target label distribution as the only unknown parameter:

$$\underbrace{p_t(f(x) = i)}_{\text{Estimated with } \mathcal{D}^t} = \sum_{j=1}^{K} \underbrace{p_s(f(x) = i, y = j)}_{\text{Estimated with } \mathcal{D}^s} \cdot \underbrace{p_t(y = j)}_{\text{Objective}} / \underbrace{p_s(y = j)}_{\text{Estimated with } \mathcal{D}^s} \cdot \tag{2.27}$$

Based on Eq. (2.27), BBSE obtains an estimate of the target label distribution $p_t(y = \cdot) = \boldsymbol{\pi}$ by solving the linear system of K equations:

$$\hat{\boldsymbol{b}} = \hat{\boldsymbol{C}}\boldsymbol{w}, \tag{2.28}$$

where 1) $\hat{\boldsymbol{b}}$ is an estimate of the parameter of the target domain marginal distribution $p_t(f(x) = \cdot) = \boldsymbol{b}$ based on the target unlabeled dataset $\mathcal{D}^t$, 2) $\hat{\boldsymbol{C}}$ is an estimate of the parameter of the source domain joint distribution $p_s(f(x) = i, y = j) = C_{ij}$ given the source labeled dataset $\mathcal{D}^s$ (*e.g.* source domain confusion matrix) and 3) $\boldsymbol{w} = p_t(y = \cdot)/p_s(y = \cdot)$ is the target over source label distribution ratio.

RLLS: Azizzadenesheli et al. (2018) proposed the Regularized Learning under Label Shift (RLLS) model based on BBSE. With the reparameterization of the parameter $\boldsymbol{w} := \mathbf{1} + \lambda\boldsymbol{\theta}$, the RLLS model optimizes the BBSE objective over $\boldsymbol{\theta} \in \mathbb{R}^K$ with an extra regularization term $\delta \cdot \|\boldsymbol{\theta}\|_2$:

$$\theta = \arg\min_{\theta} \|\hat{\boldsymbol{C}}(\lambda\boldsymbol{\theta} + \mathbf{1}) - \hat{\boldsymbol{b}}\|_2 + \delta \cdot \|\boldsymbol{\theta}\|_2, \tag{2.29}$$

where $\|\cdot\|_2$ denotes the L2-norm – the square root of the sum of the squared elements of the given vector, $\mathbf{1} = [1, 1, ..., 1]^T \in \mathbb{R}^K$ is the K-dimensional vector with 1 in all dimensions.

IWDAN: Tachet des Combes et al. (2020) added non-negative constraints to the optimisation objective of BBSE to stabilize the computation. The optimisation objective of the IWDAN model can be written as:

$$\begin{cases} \min_{\boldsymbol{w}} \|\hat{\boldsymbol{C}}\boldsymbol{w} - \hat{\boldsymbol{b}}\|_2, \\ \text{s.t: } \boldsymbol{w} \in \mathbb{R}^K_{>0} \quad \text{and} \quad \mathbf{1}^T\hat{\boldsymbol{C}}\boldsymbol{w} = 1. \end{cases} \tag{2.30}$$

MLLS: Saerens et al. (2002) proposed the MLLS model by analyse the label shift estimation problem in a simple probabilistic perspective. Under the assumption that the classifier f reflects the true conditional probability $f(x)_j = p_s(y = j|x)$, MLLS obtains a Maximum Likelihood estimate (MLE) of the target label distribution $p_t(y = \cdot) = \boldsymbol{\pi}$ through maximising the log likelihood:

$$\boldsymbol{\pi} = \arg\max_{\boldsymbol{\pi} \in \Delta^{K-1}} \log \prod_{i=1}^{N^t} p_t(x_i^t; \boldsymbol{\pi}) = \arg\max_{\boldsymbol{\pi} \in \Delta^{K-1}} \log \left(\prod_{i=1}^{N^t} \sum_{j=1}^{K} \frac{\pi_j}{c_j} f(x_i^t)_j \right) + Const. \tag{2.31}$$

The optimisation objective can be obtained via an EM algorithm, which is carried out by iterating over two steps – an E-step followed by an M-step:

$$\text{E-step:} \quad g_{ij}^{(m)} = \frac{\frac{\pi_j}{c_j} f(x_i^t)_j}{\sum_{l=1}^{K} \frac{\pi_l}{c_l} f(x_i^t)_l} \quad \text{and} \quad \text{M-step:} \quad \pi_j^{(m+1)} = \frac{1}{N} \sum_{i=1}^{N} g_{ij}^{(m)}, \tag{2.32}$$

where c_j denotes the source domain label distribution $p_s(y = j) = c_j$.

The MLLS method enjoys several theoretical and empirical advantages:

1. The EM algorithm 2.32 converges to a MLE of the target label distribution (Alexandari et al., 2020);
2. MLLS is consistent under the same consistency condition of BBSE when f is canonically calibrated on the source domain data distribution $p_s(x,y)$ (Garg, Wu, Balakrishnan et al., 2020).
3. When the NN classifier f is adjusted with a NN calibration method like Temperature Scaling or Vector Scaling (Guo, Pleiss et al., 2017), MLLS usually achieves SOTA performance (Alexandari et al., 2020);

LTF: Guo, Gong et al. (2020) leverages the flexibility of the Neural Network models to tackle the Closed Set Label Shift estimation task. In their proposed model, two conditional generative adversarial network (cGAN) (Goodfellow, Pouget-Abadie et al., 2014) are trained.

The first cGAN model is trained with the source domain labelled dataset, where the generator learns to model the source domain conditional distribution $p_s(x|y)$ and the discriminator distinguishes the source domain ground truth data-label pairs and generated pairs.

The second cGAM model is trained with the target domain unlabeled dataset, and the network that predicts the target label distribution is trained in the generator obtained from the first cGAN model (Assumption 1:$p_s(x|y) = p_t(x|y)$), and the discriminator distinguish the target domain unlabeled data from the generated unlabeled data.

ELSA: Tian, Zhang et al. (2023) utilizes the semi-parametric theory in the label shift estimation problem, where the authors construct "the perpendicular space that corresponds to the influence functions for estimating the parameter $\boldsymbol{w}$", *i.e.* the target over source label distribution ratio. A moment matching approach is then proposed based on the equality under the label shift assumption (Assumption 1):

$$\mathbb{E}_{X_s}[w_i \cdot f(X_s)_i] = \mathbb{E}_{X_t}[f(X_t)_i] \quad \text{for all} \quad i \in \{1,2,...,K-1\}, \tag{2.33}$$

where $f : \mathcal{X} \to \mathbb{R}^{K-1}$ outputs a $K-1$ dimensional real vector rather than a discrete value required in the BBSE/RLLS/IWDAN model or a probability simplex used in the MLLS model.

2.3.1.3 Closed Set Label Shift Correction

Offline Methods: Few offline label shift correction methods have been proposed. The most popular approach (Saerens et al., 2002) constructs the target domain classifier $g : \mathcal{X} \to \Delta^{K-1}$ based on the source domain classifier $f : \mathcal{X} \to \Delta^{K-1}$ and the source and target domain label distributions $p_s(y=i) = c_i$ and $p_t(y=i) = \pi_i$:

$$g(x)_j = \frac{\frac{\pi_j}{c_j} f(x)_j}{\sum_{l=1}^{K} \frac{\pi_l}{c_l} f_0(x)_l}. \tag{2.34}$$

We denote this approach as LSC. LSC has several excellent properties: 1) if $\arg\max f$ is a source domain Bayes optimal classifier implies, then $\arg\max g$ is a target domain

Problem Setup	Existing/Our Works	Approach
Closed Set Label Shift		
Detection	BBSD (Lipton et al., 2018)	Hypothesis test
Estimation	BBSE (Lipton et al., 2018)	linear system solving
	RLLS (Azizzadenesheli et al., 2018)	linear system solving
	IWDAN (Tachet des Combes et al., 2020)	linear system solving
	LTF (Guo, Gong et al., 2020)	NN model (GANs)
	MLLS (Saerens et al., 2002)	EM algorithm
	MAPLS (Chapter 3), GLSE (Chapter 4)	EM algorithm
	ELSA (Tian, Zhang et al., 2023)	fix-point iteration
Correction	LSC (Saerens et al., 2002)	prediction adjustment
	Re-weighting, Re-sampling (Section 2.3.2)	loss/sampler/logit adjustment
Open Set Label Shift		
Estimation	PULSE (Garg, Balakrishnan et al., 2022)	re-training + PU learning
	OSLS-MLE/MAP (Chapter 5)	EM algorithm
Other More Advanced/Challenging Settings		
Online LS	closed set: Wu, Guo, Su et al. (2021)	loss + gradient adjustment
	closed set: Baby et al. (2024)	re-sampling + online regression
	open set: HANOL (Qian et al., 2023)	ensemble
Active Learning	MALLS (Zhao et al., 2021)	loss + sampler adjustment
Adversarial LS	Zhang, Menon et al. (2021)	loss + gradient adjustment
Supervised LS	Maity et al. (2022)	error analysis (theoretical)
Relaxed LS	RLSbench (Garg, Erickson et al., 2023)	online&offline correction
Zero-Shot Learning	$\mathcal{L}_{BT}$ + GP (Chapter 6)	loss + Gaussian Process

Table 2.6: Summary of recent works on the Label Shift problems, including label shift *detection*, *estimation*, *correction* in the closed set classification task and other more challenging settings.

Bayes optimal classifier (Tian, Liu et al., 2020) and 2) if f is canonically calibrated on the source domain $p_s(x,y)$, then g is canonically calibrated on the target domain $p_t(x,y)$ (Podkopaev and Ramdas, 2021).

Online Methods: The online label shift correction methods are usually simple extensions of the techniques used in the class-imbalance problem. The main difference is that works in the class imbalance problem typically assume that the target domain has uniform label distribution. In contrast, existing label shift correction methods use the estimated target label distribution obtained from a label shift estimation model.

Given the target label distribution estimated with the label shift estimation model, either the loss or the data sampler is adjusted during the training phase to ensure the adjusted empirical risk (expected loss) on the source domain distribution is close to the naive empirical risk on the target domain distribution (Garg, Balakrishnan et al., 2022). These methods include: 1) **Re-weighting**: Adjust the loss of each sample that corresponds to class j with w_j, 2) **Re-sampling**: Adjust the data sampler so that sample frequencies for each class match the target label distribution $p_t(y=\cdot)=\boldsymbol{\pi}$ and 2) **Hybrid methods**: A mixture of the re-weighting, re-sampling and offline methods. Details of these methods and related literature are provided in Section 2.3.2.

2.3.1.4 Other Challenging Label Shift Problem Settings

Open Set Label Shift Problems: Under the unsupervised domain adaptation setting, Garg, Balakrishnan et al. (2022) explore the label shift problems in the open set classification task. In the proposed PULSE model, the Positive-Unlabeled (PU) learning method (Yao et al., 2020) is combined with the label shift estimation methods to estimate and correct Open Set Label Shifts.

Online Label Shift (closed set): Wu, Guo, Su et al. (2021) extend BBSE to an online learning setting with a target label distribution evolving with time. The proposed model transforms the online label shift problem into an online learning problem. The online learning problem is then tackled with two online learning methods, namely Online Gradient Descent (OGD) and Follow The History (FTH). Baby et al. (2024) also transfers the online label shift problem into an online regression problem. The proposed model utilizes the existing label shift estimation method BBSE, online regression methods and re-sampling methods (see Section 2.3.2) to obtain a better classifier.

Online Label Shift (open set): Qian et al. (2023) investigate the online label shift setting under an open set classification task, where samples from a novel class appear in the target domain. The proposed model utilizes the Closed Set Label Shift estimation model BBSE and the mixture proportion estimation method BBE (Garg, Wu, Smola et al., 2021) to estimate ID class label distribution and OOD data ratio, respectively. The deep ensemble method is further employed to stabilize the estimation result.

Active Learning Under Label Shift (closed set): Zhao et al. (2021) analyse the active learning problem under label shift, where the label distributions are different between the warm-up dataset, the unlabeled train dataset and the test dataset. To tackle this problem, the authors introduce a novel model combining the re-weighting and the re-sampling (see Section 2.3.2) approach.

Adversarial Label Shift (closed set) Zhang, Menon et al. (2021) tackles the Closed Set Label Shift problem where even the target unlabeled dataset $\mathcal{D}^t$ is not available in the problem setup (Definition 1). Based on the distributionally robust optimisation (DRO) method (Shapiro et al., 2021), a training scheme is proposed to minimise the loss over a family of possible target label distributions and convergence conditions of the method are then discussed.

Supervised/Unsupervised Label Shift (closed set) Maity et al. (2022) analyse the label shift problem in a more theoretical perspective, where the standard Closed Set Label Shift problem setup (Definition 1), and the relaxed case where an additional target labeled dataset is available are discussed. The error bounds of a label shift model for both problem setups are provided, given the model space and the learning algorithm.

Relaxed Label Shift (closed set): Garg, Erickson et al. (2023) considers the label shift problem under the setting that the label shift assumption (Assumption 1) does not hold perfectly. The authors raise several new benchmark datasets with relaxed label shifts and observe that existing domain adaptation methods usually exhibit unsatisfactory performance on these datasets. Under the unsupervised domain adaptation setup, they

propose a hybrid model that combines label shift estimation and online and offline label shift correction methods.

2.3.1.5 Comparison among existing Closed/Open Set Label Shift methods

As discussed in the previous sub-sections, the closed set label shift models BBSE (Lipton et al., 2018) and MLLS (Saerens et al., 2002) usually serve as the basis of other existing label shift models. The closed set label shift model RLLS (Azizzadenesheli et al., 2018) and IWGAN (Tachet des Combes et al., 2020) introduce regularization in the BBSE method to improve robustness. Open set label shift model PULSE (Garg, Balakrishnan et al., 2022) and online label shift model Baby et al. (2024) are also built over the BBSE model.

On the other hand, the label shift models proposed in this thesis are mainly inspired by the MLLS model. Compared with the BBSE model, the MLLS model relies on an extra assumption on the source domain classifier (Assumption 3) and expect the source domain classifier to output a probability simplex. Garg, Wu, Balakrishnan et al. (2020) argues that Assumption 3 could be satisfied if the source domain classifier is well calibrated.

Apart from models following BBSE and MLLS, LTF (Guo, Gong et al., 2020) utilize neural network models for Closed Set Label Shift estimation task and thus lacks consistency guarantees compared with BBSE and MLLS (Garg, Wu, Balakrishnan et al., 2020). ELSA (Tian, Zhang et al., 2023) demonstrates superior performance over BBSE/MLLS on CIFAR10/100 datasets. However, since they have no official implementations publicly available, the advantages and limitations of this model require further analysis.

In practice, the MLLS model is preferred when model calibration methods like temperature scaling or MixUp (Zhang, Cisse et al., 2018) are used when training the source domain classifier (Alexandari et al., 2020). Otherwise, the BBSE/RLLS/IWGAN model is preferred. A comparison of the existing Closed/Open Set Label Shift models are provided in Tab. 2.7. A comparison of existing closed/open set label shift models, our proposed Closed Set Label Shift model MAPLS (Chapter 3), GLSE (Chapter 4) and our proposed Open Set Label Shift model OSLS-MLE/MAP (Chapter 5) are provided in Tab 7.1, Chapter 7 of this thesis.

Model	MLLS	BBSE	RLLS	IWGAN	PULSE
Requires Assumption 3	✓	✗	✗	✗	✗
Close Set Estimation	✓	✓	✓	✓	✓
Open Set Estimation	✗	✗	✗	✗	✓
NO (re)training	✓	✓	✓	✓	✗
With Regularization	✗	✗	✓	✓	✗

Table 2.7: Difference between previous Closed/Open Set Label Shift estimation and correction models. BBSE/RLLS/IWGAN and PULSE are effective without extra assumptions on the classifier required under their corresponding label shift problem setup. However, if the classifier is further trained with calibration methods like temperature scaling, Assumption 3 may be satisfied and the MLLS model could be more effective

2.3.2 Class Imbalance Problem

We provide a brief literature review on the class imbalance problem as this task can be considered as a special case of the label shift problem (Ye et al., 2024). A more comprehensive review of the class imbalance problem can be found in Buda et al. (2018a); Ghosh et al. (2024); Johnson and Khoshgoftaar (2019); Rezvani and Wang (2023).

In classification tasks, the class imbalance problem refers to the problem that the classifier trained on a dataset with an imbalanced distribution among classes tends to predict test images in favour of the major classes (Ali et al., 2013; Barandela et al., 2003; Buda et al., 2018a; Japkowicz and Stephen, 2002). Class imbalance problems can lead to a decrease in classification performance if the source domain has an imbalanced label distribution (Buda et al., 2018b; Wang, Ramanan et al., 2017). Existing models for class imbalance problems can be categorised into roughly four groups: 1) re-sampling methods adjust the data sampler during training; 2) re-weighting methods adjust the loss during training; 3) post-processing methods adjust the prediction in test time; 4) mixture methods that combine the three methods.

Re-Sampling Methods: The re-sampling methods aim at training the classifier with class-balanced data. Early methods proposed to up-sample tail classes (Buda et al., 2018b; Byrd and Lipton, 2019) or down-sampling head classes (Buda et al., 2018b; He and Garcia, 2009).

Re-Weighting Methods: Earlier works (He and Garcia, 2009; He and Ma, 2013; Huang, Li, Loy et al., 2016) re-weight the training loss so that the empirical risk is over a class-balanced distribution. The re-weighting approach is similar to the importance-weighted ERM proposed by BBSE (Lipton et al., 2018). These methods have been shown to overfit rare classes (Chawla et al., 2002) on highly imbalanced train sets.

Mixture Methods: Cao et al. (2019) proposes a re-weighting loss named the LDAM loss. The LDAM loss is designed to minimize the generalization error bound for a classifier on a uniform test set. The combination of the LDAM loss and the re-weighting approach demonstrates significant improvement in classification performance.

2.3.3 Long-Tailed Recognition

We provide a brief literature review on the Long-Tailed recognition problem as this task is considered an extreme case of the label shift problem (Xu et al., 2021). A more comprehensive review of the Long-Tailed recognition problem can be found in Yang, Jiang et al. (2022); Zhang, Kang et al. (2023).

Long-Tailed Recognition task considers a more realistic case of the class-imbalance problem, where the source domain (train set) has highly imbalanced or even Long-Tailed label distribution (Liu, Miao, Zhan, Wang, Gong and Yu, 2019; Wu, Guo, Luo et al., 2024). In this case, the training dataset may contain thousands of samples for each major class but only a handful of samples for each rare class. The classifier trained on the Long-Tailed train dataset faces the problem of highly imbalanced label distribution and few-shot learning in tail classes (Liu, Miao, Zhan, Wang, Gong

and Stella, 2022). Based on the high-level idea, existing works on the Long-Tailed problem can be roughly separated into the following categories: 1) decision boundary balancing, 2) fine-tuning, 3) knowledge transfer, 4) data augmentation and 5) ensemble methods.

Decision Boundary Balancing: Shifting the decision boundary between head and tail classes can also bring benefits to Long-Tailed classification. Cao et al. (2019) proposed an LDAM loss to learn a better decision boundary between imbalanced classes. OT (Peng, Sun et al., 2022) introduces a novel optimal transport algorithm to optimise a classifier for a uniform test set. The proposed model adjusts the output of a classifier for optimal decision boundary. Menon et al. (2020) demonstrate that adjusting the logits of the NN model instead of the final prediction can also improve the Long-Tailed classification performance.

Two-Stage Model (Fine-Tuning): Kang et al. (2020) proposed several two-stage training models. Models are trained normally in the first stage and fine-tuned in the second stage to adjust the decision margin in favour of tail classes. These models include classifier retraining (cRT), τ-normalization and learnable weight scaling (LWS). MiSLAS (Zhong et al., 2021) proposed a model that combined cRT and LWS and achieved better performance. Dong et al. (2022) utilizes the prompt tuning techniques to fine-tune the classifier.

Knowledge Transfer: Transferring knowledge from the head and medium classes to the tail classes may also help improve classification performance. m2m (Kim et al., 2020) synthesized tail class images with knowledge learned from head class images. Similar to prototypical methods (Boney and Ilin, 2017; Gao et al., 2019), OLTR (Liu, Miao, Zhan, Wang, Gong and Yu, 2019) and Zhu and Yang (2020) proposed to learn prototypes of each class and construct classifiers based on the prototypes. Du et al. (2023) construct the classifier by analyzing the relations between the ground truth classes and their corresponding super-classes. Han, Ye et al. (2024) leverages the diffusion model (Ho et al., 2020; Song et al., 2020) to transfer knowledge of head class to tail classes by generating tail class image features.

Data Augmentation: Data augmentation is a well-known strategy to improve classification performance (Han, Liang et al., 2022; Müller et al., 2019; Zhang, Cisse et al., 2018). In the Long-Tailed problem, MetaSAug (Li, Gong et al., 2021) adopts the ISA (Wang, Pan et al., 2019) method to generate new samples for tail classes. Remix (Chou, Chang et al., 2020) aims to modify labels of augmented samples to emphasize tail classes. UniMix (Xu et al., 2021) proposed to augment class-balanced samples during training. Yue, Mou et al. (2024) propose to combine a class-balanced loss with a data augmentation method to train the classifier.

Ensemble Methods: Deep Ensemble approaches have been shown to effectively improve the performance of Neural Network models Durasov et al. (2021); Wen, Tran et al. (2019). Ensemble models are proposed in many recent papers to improve long-tailed classification performance, with different ensemble members referred to as "experts". BBN (Zhou, Cui et al., 2020), LFME (Xiang et al., 2020) and ACE (Cai et al., 2021) train experts with different sub-datasets that emphasize head, medium or tail classes. RIDE (Wang, Lian et al., 2020) trains each expert with the entire dataset with a routing mechanism that improves model efficiency.

2.3.4 Out-of-Distribution Detection

We provide a brief literature review on the OOD detection models used in the Open Set Label Shift model proposed in this thesis (Chapter 5). A more comprehensive review of this problem can be found in Salehi et al. (2021); Yang, Zhou et al. (2024).

OOD detection has been widely studied in the Deep Learning regime. Existing approaches can be categorized into post-hoc inference methods and training methods with or without OOD data.

Post-hoc Inference: Most OOD methods are post-hoc inference methods, where the OOD classifier is constructed based on a pre-trained classifier over ID classes. OpenMax (Bendale and Boult, 2016) proposed to construct the OOD classifier by modelling per-class features with a Weibull distribution. MSP (Hendrycks and Gimpel, 2022) utilize the maximal SoftMax score of the ID classifier prediction. ODIN (Liang et al., 2017) observed that NN models respond to ID and OOD data differently under adversarial attacks (Goodfellow, Shlens et al., 2014). MDS (Lee et al., 2018a) also adopts the adversarial attack approach but detects OOD data with a Mahalanobis distance-based score. OpenGAN (Kong and Ramanan, 2021) trains an extra discriminator network to distinguish ID and OOD features. EBO (Liu, Wang et al., 2020) proposed an Energy-based score to detect OOD samples. GRAM (Sastry and Oore, 2020) establish their model with Gram matrices. ReAct (Sun, Guo et al., 2021) demonstrates that rectifying the penultimate layer features of the pre-trained classifier can help post-hoc OOD detection methods. MLS (Hendrycks, Basart et al., 2019a) argues that the maximal logit score is a better OOD indicator. VIM (Wang, Li et al., 2022a) propose a three-stage pipeline to compute the OOD score by adjusting the features, logits and SoftMax probability of the ID classifier. Hong, Fang et al. (2023); Hong, Li et al. (2023) utilize spherical geometry and hyperbolic geometry constraints to compute the OOD score. Sun, Ming et al. (2022) introduces a k-Nearest Neighbor (KNN) based OOD classifier. Ash (Djurisic et al., 2022) shows that pruning image features in the intermediate layers can help OOD detection.

Online Methods: Hendrycks, Mazeika et al. (2019) argues that training the classifier with an auxiliary self-supervised rotation loss is beneficial to OOD detection models. GODIN (Hsu et al., 2020) extends the ODIN model by introducing an extra linear layer that models the probability of the data being not OOD given the image. CSI (Tack et al., 2020) enhance a baseline OOD detector by training the classifier with a loss that contrasts ground truth samples with distribution-shifted samples. APRL (Chen, Peng et al., 2021) encourages ID samples to move far away from a bounded space left for OOD data.

Theoretical Analysis: In the machine learning community, Miller et al. (2021); Vaze et al. (2021) argues that a good ID classifier implies a good OOD classifier. Hein et al. (2019) shows that for the OOD sample, a ReLU network can predict its label as an ID class with arbitrary high confidence. Meinke and Hein (2019) propose a GMM-based classifier approach to prevent the model from assigning OOD data with high confidence. Fang et al. (2022) analyses the conditions under which OOD detection is learnable.

2.3.5 Zero-Shot Classification

We provide a brief literature review on the Zero-Shot classification problem as it is analyzed under the label shift problem setup in this thesis (Chapter 6). A more comprehensive review of this problem can be found in Pourpanah et al. (2022); Wang, Zheng et al. (2019); Xian, Lampert et al. (2019).

Traditional and Generalized Zero-Shot Learning (ZSL): Early ZSL research adopts a so-called Traditional ZSL setting (Akata et al., 2015; Norouzi et al., 2013). The Traditional ZSL requires the model to train on images of seen classes and semantic vectors of seen and unseen classes. Test images are restricted to the unseen classes. However, in practice, test images may also come from the seen classes (Xian, Lampert et al., 2019). The Generalized ZSL setting was proposed to address the problem of including both seen and unseen images in the test set. According to Xian, Lampert et al. (2019), models that perform well in the Traditional ZSL setting may not work well in the Generalized ZSL setting.

In recent years, ZSL has been explored with the help of the large language models or large vision models (Kojima et al., 2022; Törnberg, 2023). These works usually have different problem setups compare with the Zero-Shot classification problem considered in this thesis.

Prototypical Methods. In the prototypical methods (Boney and Ilin, 2017; Gao et al., 2019; Snell et al., 2017b), a prototype is learned for each class to help with classification. For example, Snell et al. (2017b) propose a neural network to learn a projection from semantic vectors to feature prototypes of each class. Test samples are classified via Nearest Neighbor among prototypes. While the classification process of our model is similar to prototypical methods, our model uses a Gaussian Process Regression instead of Neural Networks to predict prototypes of unseen classes.

Inductive and Transductive ZSL: Inductive ZSL requires that no feature information of unseen classes is present during the training phase (Xian, Lampert et al., 2019). Models that introduce unlabeled unseen images during the training phase are called transductive ZSL models (Chapelle et al., 2009). Ensuring a fair comparison, results from such models are usually compared separately to inductive models since additional information is introduced (Li, Min et al., 2019; Meng and Guo, 2017; Verma and Rai, 2017).

Other Zero-Shot Learning Problems: In recent years, the Zero-Shot learning task has been explored in the more challenging problem settings like ZSL in the real world large-scaled datasets (Radford et al., 2021), using audio information instead of semantic information in the ZSL task (Hong, Hayder et al., 2023; Zheng et al., 2023) or Zero-Shot domain adaptation tasks (Kutbi et al., 2021; Peng, Wu et al., 2018).

2.4 Summary

In this chapter, we provide the formal definitions of the label shift problems that we consider, briefly introduce the background methods that will be used in the later chapters to solve the label shift problems and review the literature in the related research fields. In the following chapters, we will discuss the label shift problem in

closed set, open set and zero-shot classification setups, with the help of the Machine Learning/Deep Learning methods provided in the background section.

Chapter 3

Classifier Based Closed Set Label Shift

3.1 Introduction

This chapter focuses on the Closed Set Label Shift (CSLS) estimation and correction problem. We aim to develop a label shift estimation model that is robust for real world datasets, which usually have 1) a large number of classes and 2) a highly imbalanced or even Long-Tailed label distribution on the source domain train set (Liu, Miao, Zhan, Wang, Gong and Yu, 2019).

Refresher on the CSLS problem In *Closed Set Label Shift* problems, we are given a source domain labeled dataset, a target domain unlabeled dataset and a source domain classifier. The objectives include 1) *detection*: verify if the label distributions between the source and target domain are identical, 2) *estimation*: estimate the target label distribution and 3) *correction*: construct an appropriate target domain classifier. The mathematical definition is given in the next section (duplicating the original given in Chapter 2).

Motivations Two observations motivate us to investigate the CSLS problem in this chapter. Firstly, real world classification problems usually have large-scale datasets with large numbers of classes or highly imbalanced label distributions. Hence, the robustness of the label shift models on these datasets should be taken into account. Secondly, existing Closed Set Label Shift estimation models are usually examined on small-scale datasets like MNIST/CIFAR10. The performance of these models on large-scale datasets is rarely discussed.

Contributions In this chapter, we propose a novel CSLS estimation model under a Bayesian framework. We construct the Bayesian posterior of the target label distribution parameters given data and a prior. We derive a novel EM algorithm to obtain an MAP estimate of the target label distribution. We further propose 1) a novel Adaptive Prior Learning (APL) model that adaptively chooses the prior parameters given data, and 2) and a posterior sampling model that uses MCMC to draw i.i.d. samples from the posterior. To the best of our knowledge, Bayesian analysis has never been used in previous label shift estimation works.

Extensive experiments are conducted with different types of Closed Set Label Shifts. In contrast to previous methods that mainly focus on MNIST and CIFAR10, the

proposed model is evaluated on the CIFAR100, ImageNet, Places datasets and Long-Tailed versions of each dataset. For target label distributions, as well as evaluating under previous label shift estimation settings (Lipton et al., 2018) with Dirichlet shift, we also introduce Long-Tailed benchmark test set shifts proposed in Long-Tailed classification (Hong, Han et al., 2021). Experimental results show that our model consistently outperforms existing state-of-the-art (SOTA) models, particularly obtaining better accuracy when the train set is highly imbalanced. These results demonstrate the applicability of our model to real world label shift tasks.

The contributions of this chapter are as follows:

1. A novel classifier-based label shift model is proposed under a Bayesian framework. The proposed model estimates and corrects label shift without retraining the classifier. The posterior of the target label distribution given data and a prior is constructed.
2. A novel EM algorithm that computes the maxima of the posterior (MAP estimate) is derived, which minimizes a strictly convex objective. A novel Adaptive Prior Learning (APL) model is proposed to determine the parameters of the prior adaptively given data.
3. A novel posterior sampling model is proposed to estimate and correct label shift based on i.i.d. samples drawn from the posterior via MCMC.
4. Experiments show that the proposed model consistently outperforms previous label shift estimation models in various label shift settings on CIFAR100, ImageNet, Places and the Long-Tailed version of each dataset.

Background and Related Works Background about the Bayesian Inference methods (including MAP estimate and MCMC methods) can be found in Chapter 2, Section 2.2.3. Background of the EM algorithms can be found in Chapter 2, Section 2.2.5. The related literature on the CSLS problem, the Class Imbalance Problem and the Long-Tailed Recognition problem are discussed in Chapter 2, Section 2.3.1, Section 2.3.2 and Section 2.3.3, respectively.

3.2 Problem Setup and Analysis

3.2.1 Definition and Assumptions

In the Closed Set Label Shift (CSLS) problem, recall that in Chapter 2, Assumption 1, we have the Closed Set Label Shift assumption available:

Assumption 1. (Closed Set Label Shift Assumption)

$$p_s(x|y=i) = p_t(x|y=i) \quad \text{for all} \quad i \in \mathcal{Y}. \tag{2.1}$$

Under Assumption 1, we are given a labeled source domain dataset $\mathcal{D}^s$, an unlabeled target domain dataset $\mathcal{D}^t$ and a source domain classifier f. The *detection*, *estimation* and *correction* problems are then defined as:

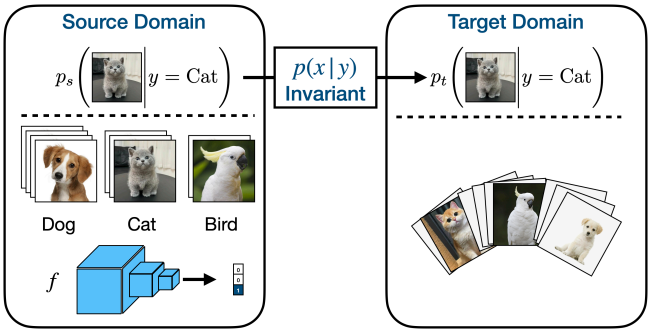

Figure 3.1: Closed Set Label Shift problem setup, where we are given: 1) **assumption**: the Closed Set Label Shift assumption (Assumption 1), 2) **datasets**: a source domain labeled dataset and a target domain unlabeled dataset and 3) **model**: a source domain classifier or feature extractor f. This information is used for closed label shift ***detection***, ***estimation*** and ***correction*** problems.

Definition 1. (**Closed Set Label Shift Problem**)

Under Assumption 1, given:

- Source domain labeled data $\mathcal{D}^s = \{(x_i^s, y_i^s)\}_{i=1}^{N^s}$ (K classes);
- Target domain unlabeled data $\mathcal{D}^t = \{x_i^t\}_{i=1}^{N^t}$ (K classes);
- Source domain classifier f.

the CSLS problem is to solve

- *Detection*: Verify $p_s(y = \cdot) = p_t(y = \cdot)$;
- *Estimation:* Estimate $p_t(y = \cdot)$;
- *Correction:* Model $p_t(y = \cdot|x)$ based on f.

A graphical model depiction of the feature based CSLS estimation problem is given in Figure 3.2. This chapter focuses on the CSLS estimation and correction problems, where our objective is to 1) estimate the target ID label distribution $\boldsymbol{\pi}$ and 2) use these estimates to build a better target domain classifier.

In Definition 1, the feature extractor f can be obtained by training a NN model on the source domain dataset via supervised learning. This chapter considers the classifier that outputs SoftMax probabilities (probability simplex), where we assume that the classifier f models the conditional distribution $p_s(y = \cdot|y)$:

Assumption 3. (**CSLS Classifier Based Approach Assumption**)

$$p_s(y = i|x) = f(x)_i \quad \text{for all} \quad i \in \mathcal{Y}. \tag{3.1}$$

We superimpose our assumptions onto the graphical model in Figure 3.2.

Validity of the Assumption: Assumption 3 is a common assumption used in the Deep Learning literature, including label shift estimation problem (Ye et al., 2024), model calibration (Liang et al., 2017) and Long-Tailed Recognition (Xu et al., 2021).

According to Garg, Wu, Balakrishnan et al. (2020), if Assumption 3 is satisfied, classifier f is a perfectly calibrated classifier on the source domain. In practice, Alexandari et al. (2020) observed that if calibration methods like temperature scaling are used, the MLLS model could outperform other previous models. This observation implies that calibration performance metrics like Expected Calibration Error (ECE) (Guo, Pleiss et al., 2017; Liu, Ye, Cui et al., 2024; Liu, Ye, Wang et al., 2023) can be used to judge if Assumption 3 is satisfied. Therefore when ECE of the classifier is low on the source domain dataset, the MLLS model could be more effective.

3.2.2 Graphical Model Setup

Observing that RVs of the source and target domain label Y_s, Y_t are supported on a finite and discrete set $\mathcal{Y} = \{1, 2, ..., K\}$ with each element $i \in \mathcal{Y}$ represents an independent class, thus Y_s, Y_t follow categorical distributions $\text{Cat}(K, \cdot)$ over K classes. Here we denote $Y_s \sim \text{Cat}(K, \boldsymbol{c})$ and $Y_t \sim \text{Cat}(K, \boldsymbol{\pi})$. The parameters $\boldsymbol{c}, \boldsymbol{\pi}$ of categorical distributions are K dimensional probability simplexes that satisfy $p_s(y = i) = c_i$ and $p_t(y = i) = \pi_i$ for all $i \in \mathcal{Y}$. Thus we have $\boldsymbol{c}, \boldsymbol{\pi} \in \Delta^{K-1}$ where Δ^{K-1} denotes the space of a K dimensional probability simplex. Estimating the target label distribution $p_t(y)$ is equivalent to estimating parameters $\boldsymbol{\pi}$ in $\text{Cat}(K, \boldsymbol{\pi})$.

The graphical model setup along with our assumptions is illustrated in Fig. 3.2. We employ a prior (with parameter $\boldsymbol{\alpha}$) over the parameter $\boldsymbol{\pi}$ of the target label distribution $p_t(y = \cdot) = \boldsymbol{\pi}$, which is further discussed in subsequent sections. We treat the parameter $\boldsymbol{c}$ of the source label distribution $p_s(y = \cdot) = \boldsymbol{c}$ as deterministic parameters, which is directly estimated with the source domain labeled dataset $\mathcal{D}^s$.

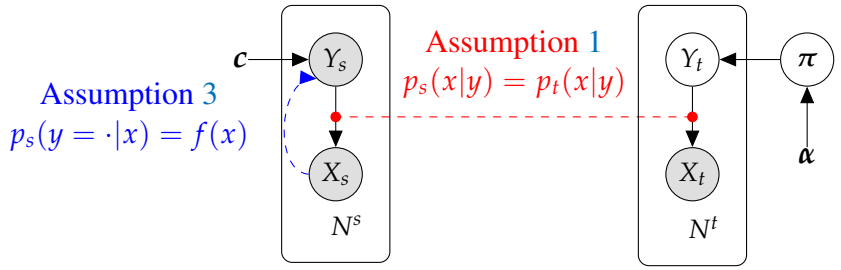

Figure 3.2: Graphical model of the classifier based Closed Set Label Shift setting and our assumptions. X_s, X_t are data for the source and target domain, Y_s, Y_t are the corresponding categorical-valued labels. $\boldsymbol{c}, \boldsymbol{\pi}$ are source and target domain label distribution class probabilities. Source domain data X_s is observed with ground truth ID data in $\mathcal{D}^s$. $p(x|y)$ are invariant under the label shift assumption. We estimate $\boldsymbol{\pi}$ by assuming the classifier f reflects the posterior $p_s(y|x)$ (Assumption 3).

3.3 Proposed Method

3.3.1 Model Overview

This section introduces a classifier based model for the Closed Set Label Shift estimation problem (estimating the target label distribution $p_t(y = \cdot) = \boldsymbol{\pi}$).

Method Summary: The main idea of the model proposed in this chapter is to introduce a regularization mechanism in the CSLS estimation model to improve the robustness of the model under large label shift between the source and the target domain.

This objective is achieved by utilizing the Bayesian approach, where the regularization term is introduced through employing a prior distribution over the estimation objective. Specifically, by employing a prior distribution over the parameter of the target label distribution $p_t(y = \cdot) = \boldsymbol{\pi}$, we can construct the posterior of $\boldsymbol{\pi}$. The MAP estimate of $\boldsymbol{\pi}$ can then be obtained by deriving an EM algorithm (background Section 2.2.5) to find the maximal point of the posterior, which can serve as an estimate of the target label distribution. The i.i.d. samples of the posterior can also be obtained through an MCMC sampling method, which further provide the uncertainty information of the estimation.

Content Outline:

1. (Section 3.3.2) Construct the posterior of the parameter $\boldsymbol{\pi}$ of the target label distribution $p_t(y = \cdot) = \boldsymbol{\pi}$.
2. (Section 3.3.3) Obtain the MAP estimate of $\boldsymbol{\pi}$ by maximizing the posterior with an EM algorithm.
3. (Section 3.3.4) Determine prior parameter $\boldsymbol{\alpha}$ that is required in the posterior.
4. (Section 3.3.5) Obtain i.i.d. samples from the posterior via MCMC if uncertainty is of interest.
5. (Section 3.3.6) Discuss the choices to estimate source domain label distribution required in the posterior.
6. (Section 3.3.7) Summarizes the overall classifier based label shift estimation and correction model.

All the theoretical proofs in our model are available in Appendix A.1.

3.3.2 Negative Log Posterior

We propose a novel Bayesian approach for the label shift estimation problem. By employing a prior distribution over target label distribution $p_t(y = \cdot) = \boldsymbol{\pi}$, we obtain the posterior of $\boldsymbol{\pi}$ given available data $\mathcal{D}^t$. Based on the posterior, we derive an EM algorithm to obtain the Maximum *A Posteriori* (MAP) estimate of $\boldsymbol{\pi}$. To utilize the information of the entire posterior, we also propose to use Hamiltonian Monte-Carlo (HMC) method to obtain i.i.d. samples from the posterior.

The categorical distribution $Y_t \sim \mathrm{Cat}(K, \boldsymbol{\pi})$ requires that the prior distribution over $\boldsymbol{\pi}$ is supported on Δ^{K-1}. K dimensional Dirichlet distributions satisfy this constraint, and are often used as a prior over parameters of categorical distributions (Joo et al., 2020; Tu, 2014). Therefore, we employ a Dirichlet prior over the parameters $\boldsymbol{\pi} \sim \mathrm{Dir}(K, \boldsymbol{\alpha})$ of the target label distribution $\mathrm{Cat}(K, \boldsymbol{\pi})$, where $\boldsymbol{\alpha} \in \mathbb{R}^K_{>1} = \{x \in \mathbb{R}^K | x > 1\}$. With the Dirichlet prior as $p(\boldsymbol{\pi}|\boldsymbol{\alpha})$ and unlabeled target domain samples $\mathcal{D}^t$, we can write

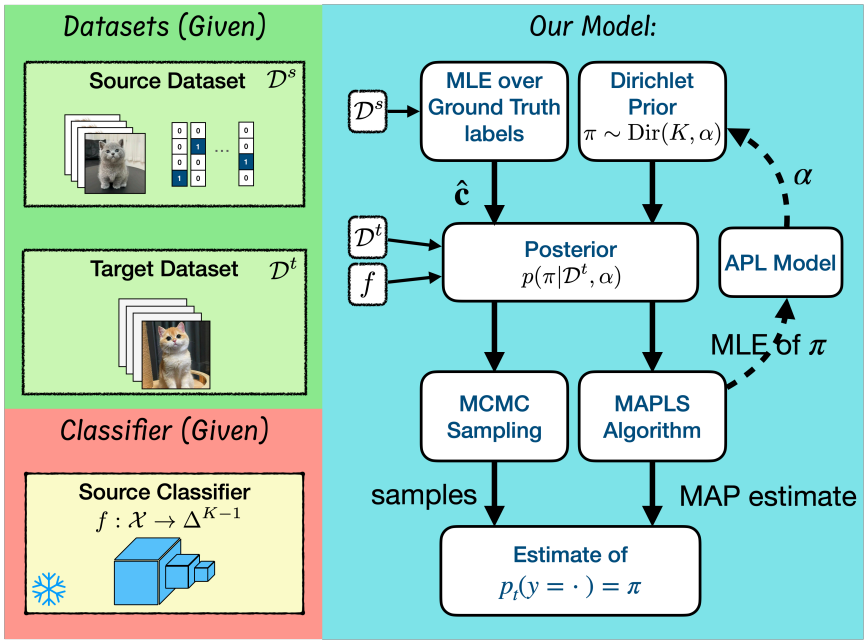

Figure 3.3: Structure of our proposed Closed Set Label Shift estimation model. We construct the analytical Bayesian posterior of the target label distribution given the target domain dataset and a Dirichlet prior. Then, based on our proposed APL model that adaptively learns prior parameters $\boldsymbol{\alpha}$ given data, we derive a MAPLS algorithm to obtain a MAP estimate of $\boldsymbol{\pi}$ and propose a posterior sampling model that uses MCMC to obtain samples from the posterior.

the posterior of $\boldsymbol{\pi}$ given $\mathcal{D}^t$ and $\boldsymbol{\alpha}$ as:

$$p(\boldsymbol{\pi}|\mathcal{D}^t, \boldsymbol{\alpha}) = \frac{1}{Z} p(\boldsymbol{\pi}|\boldsymbol{\alpha}) \prod_{i=1}^{N^t} p_t(x_i^t|\boldsymbol{\pi}), \tag{3.2}$$

where $Z = \int p(\mathcal{D}^t|\boldsymbol{\pi}')p(\boldsymbol{\pi}'|\boldsymbol{\alpha}))d\boldsymbol{\pi}'$ is a constant w.r.t. $\boldsymbol{\pi}$.

The marginal distribution $p_t(x|\boldsymbol{\pi})$ can be rewritten as a combination of known expressions. Given the source domain labeled data, we can estimate the source domain label distribution $p_s(y = j) = c_j$ in $Y_s \sim \mathrm{Cat}(K, \boldsymbol{c})$, which is also a categorical distribution. $p_s(y = j|x_i^t) = f(x_i^t)_j$ for all data $x_i^t \in \mathcal{D}^t$ under Assumption 3, and the target label distribution is $p_t(y = j) = \pi_j$. Formally we are given:

$$p_s(y = j) = c_j > 0, \qquad p_s(y = j|x_i^t) = f(x_i^t)_j \quad \text{and} \quad p_t(y = j) = \pi_j, \tag{3.3}$$

where $c_i > 0, i = 1, 2...K$ because each class has non-zero sample frequency on the source domain.

With Eq. (3.3) available, utilizing Bayes rule, we can rewrite the posterior in Eq. (3.2) as:

$$p(\boldsymbol{\pi}|\mathcal{D}^t, \boldsymbol{\alpha}) = \frac{1}{Z} p(\boldsymbol{\pi}|\boldsymbol{\alpha}) \prod_{i=1}^{N^t} \sum_{j=1}^{K} p_t(x_i^t) \frac{\pi_j}{c_j} f(x_i^t)_j. \tag{3.4}$$

Note that $p_t(x_i^t)$ and Z are constants w.r.t. $\boldsymbol{\pi}$ and $p(\boldsymbol{\pi}|\boldsymbol{\alpha})$ is the Dirichlet prior. Therefore the analytical expression for the un-normalized posterior $p(\boldsymbol{\pi}|\mathcal{D}^t, \boldsymbol{\alpha})$ can

be obtained from Eq. (3.4).

3.3.3 Maximum *a Posteriori* estimate

We first derive an EM algorithm to obtain MAP estimate of $\boldsymbol{\pi}$. By definition, any MAP estimate $\boldsymbol{\pi}^*$ minimizes the negative log posterior:

$$\boldsymbol{\pi}^* \in \arg\min_{\boldsymbol{\pi} \in \Delta^{K-1}} -\log p(\boldsymbol{\pi}|\mathcal{D}^t, \boldsymbol{\alpha}) \tag{3.5}$$

We prove that the optimization problem defined in Eq. (3.5) is strictly convex in $\boldsymbol{\pi}$ and propose a novel EM algorithm to find $\boldsymbol{\pi}^*$. We name our proposed algorithm: Maximum *a Posteriori* Label Shift (MAPLS).

Proposition 1. *Under Assumption 1 and Assumption 3. Let* $\boldsymbol{\pi} \sim Dir(K, \boldsymbol{\alpha})$ *with* $\boldsymbol{\alpha} \in \mathbb{R}^K_{>1}$. *Then in Eq. (3.5), the objective is strictly convex in* $\boldsymbol{\pi}$, $\boldsymbol{\pi}^*$ *is unique and EM Algorithm 1 converges to* $\boldsymbol{\pi}^*$.

Algorithm 1 MAPLS

Input:

- Target domain unlabeled data $\mathcal{D}^t = \{x_i^t\}_{i=1}^{N^t}$;
- Parameter of source domain label distribution: $p_s(y = j) = c_j$;
- Source domain classifier $f(x)$;
- Dirichlet prior $p(\boldsymbol{\pi}|\boldsymbol{\alpha})$ with $\boldsymbol{\alpha} \in \mathbb{R}^K_{>1}$.

Initialize: $\boldsymbol{\pi}^{(0)} \in \Delta^{K-1}_{>0}$.

for $t = 0$ to M **do**

E-step Evaluate $g_{ij}^{(m)}$:

$$g_{ij}^{(m)} = \frac{\frac{\pi_j^{(m)}}{c_j} f(x_i^t)_j}{\sum_{l=1}^K \frac{\pi_l^{(m)}}{c_l} f(x_i^t)_l}. \tag{3.6}$$

M-step Obtain $\boldsymbol{\pi}^{(m+1)}$ with:

$$\pi_j^{(m+1)} = \frac{\sum_{i=1}^{N^t} g_{ij}^{(m)} + \alpha_j - 1}{N^t + \sum_{l=1}^K (\alpha_l - 1)}. \tag{3.7}$$

end for

Output: $p_t(y = \cdot) = \boldsymbol{\pi}^{(m+1)}$

The detailed proof can be found in Appendix A.1.1, A.1.2.

Algorithm 1 can seen as a generalization of MLLS. In the M-Step, we can rewrite Eq. (3.7) as:

$$\pi_j^{(m+1)} = \lambda \underbrace{\frac{\sum_{i=1}^{N^t} g_{ij}^{(m)}}{N^t}}_{\text{Data contribution}} + (1-\lambda) \underbrace{\frac{\alpha_j - 1}{\sum_{l=1}^K (\alpha_l - 1)}}_{\text{Prior contribution}} \tag{3.8}$$

where $\lambda \in (0,1)$ has the form:

$$\lambda = \frac{N^t}{N^t + \sum_{l=1}^{K}(\alpha_l - 1)}. \tag{3.9}$$

As $\lambda \to 1^-$, the algorithm degenerates to MLLS. As $\lambda \to 0^+$, the MAP estimate will converge to the Dirichlet prior $\mathrm{Dir}(K, \boldsymbol{\alpha})$. In this manner, λ can be seen as our confidence in our label distribution estimation.

The choice of $\boldsymbol{\alpha}$ and corresponding λ affect the MAP estimate $\boldsymbol{\pi}^*$. In practice, it is important to determine an appropriate $\boldsymbol{\alpha}$ and λ to give a good MAP estimate $\boldsymbol{\pi}^*$ for the target label distribution $p_t(y)$.

After obtaining $\boldsymbol{\pi}^*$, we can use Eq. (2.34) to correct the source domain classifier f to the target domain under label shift.

Symmetric Dirichlet Prior: The Dirichlet prior possesses K parameters in $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_K] \in \mathbb{R}^K_{>1}$. When no information about the target domain label distribution is available, we may set $\alpha_j = \alpha_0$. This has the advantage of reducing the number of parameters to be chosen, at the cost of limiting expressivity.

- *The Dirichlet prior satisfies* $\boldsymbol{\pi} \sim Dir(K, \alpha_0 \mathbf{1})$.

Then the M-Step of the MAPLS algorithm in the form of Eq. (3.8) can be further simplified as:

$$\pi_j^{(m+1)} = \lambda \frac{\sum_{i=1}^{N^t} g_{ij}^{(m)}}{N^t} + (1-\lambda)\frac{1}{K} \tag{3.10}$$

where $\lambda = N^t / (N^t + K(\alpha_0 - 1))$ also has a simpler form.

The MAPLS algorithm with $\lambda \to 0^+$ will converge to a uniform categorical distribution with $\boldsymbol{\pi} = \mathbf{1}/K$ in $Y_t \sim \mathrm{Cat}(K, \boldsymbol{\pi})$. Note that now $\boldsymbol{\alpha} = \alpha_0 \mathbf{1}$ is fully determined by λ, and we can determine parameter α_0 in the prior by selecting a value for λ. In this case, $1 - \lambda$ represents the strength of regularization in the MAP estimation procedure.

3.3.4 Adaptive Prior Learning Model

In our MAPLS algorithm 1, the prior parameter $\boldsymbol{\alpha}$ should be determined before the estimation of $\boldsymbol{\pi}$. In this Chapter, based on the analysis of the possible estimation error, we propose a novel Adaptive Prior Learning (APL) model to adaptively learn $\boldsymbol{\alpha}$ given available data. Our model is inspired by the empirical Bayesian (Casella, 1992; Robbins, 1992) approach.

Estimation Error Analysis: Intuitively, two factors can induce estimation error in our posterior. Firstly, we use a classifier $f(x)$ to model ground truth $p_s(y|x)$ in Assumption 1, when the classifier fails to represent the ground truth, the model is subject to misspecification error. Secondly, even if Assumption 1 is satisfied, our MAPLS model will have an associated sampling error due to using a finite number of samples, like other models (Garg, Wu, Balakrishnan et al., 2020).

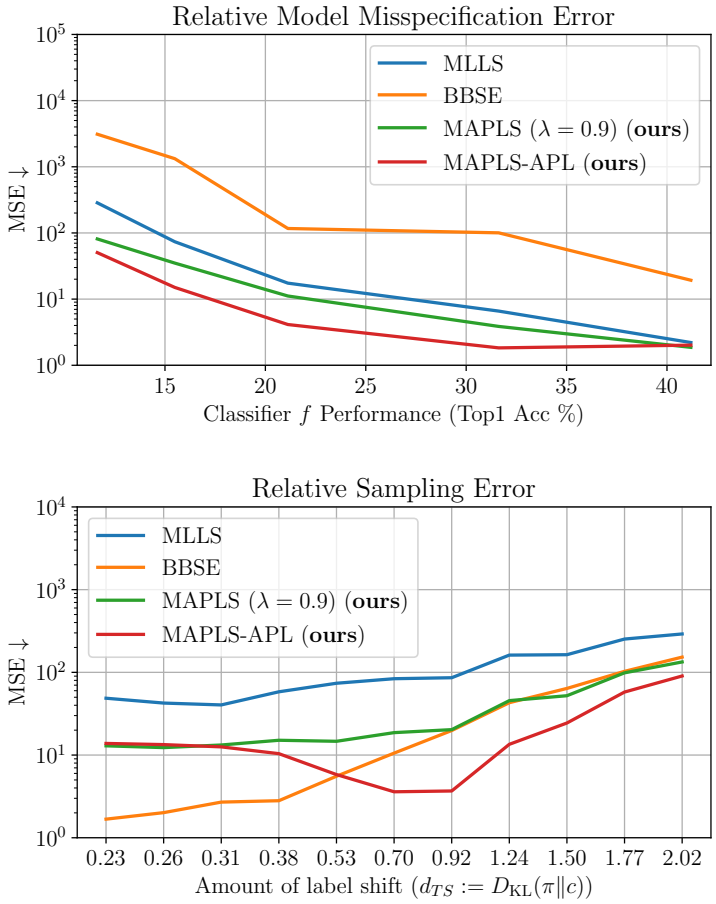

Figure 3.4: Label shift estimation error analysis. The Mean Square Error (MSE, see Section 3.4.1) increases when: (1) the model is misspecificed, *i.e.* $p_s(y) = f(x)$ is not satisfied (left); (2) the sampling error gets magnified when source and target domains have large label shift (right). Our MAPLS with fixed prior ($\lambda = 0.9$ in Eq. (3.10)) can reduce both errors compared with MLLS. Our MAPLS-APL model with prior parameters learned given data can further reduce MSE and outperform BBSE under large label shift (right).

Adaptive Prior Learning: In our APL model, we propose to use a heuristic to loosely evaluate the magnitude of model misspecification error. The sampling error of the label shift estimation model can be magnified with large label shift between the target and source domains (*e.g.* Fig. 3.4). Therefore, our APL model also includes a heuristic to mitigate sampling error.

Practically, we first run MAPLS with $\lambda = 1$ to obtain an initial MLE of the target label distribution $\boldsymbol{\pi}^{MLE}$. Then our APL model quantifies the two estimation errors based on the three KL-divergences below:

$$\begin{cases} d_{SU} := D_{\mathrm{KL}}(\boldsymbol{c} \| \mathbf{1}/K) \\ d_{TU} := D_{\mathrm{KL}}(\boldsymbol{\pi}^{MLE} \| \mathbf{1}/K) \\ d_{TS} := D_{\mathrm{KL}}(\boldsymbol{\pi}^{MLE} \| \boldsymbol{c}), \end{cases} \tag{3.11}$$

where 1) $\mathbf{1}/K$ is the parameter of a uniform categorical distribution $\mathrm{Cat}(K, \mathbf{1}/K)$ for K classes, 2) $\boldsymbol{c}$ is the parameter of the source label distribution, 3) S, T, U represents source, target and uniform label distribution respectively and 4) $D_{\mathrm{KL}}(\boldsymbol{\alpha} \| \boldsymbol{\beta})$ denotes the KL-divergence of two categorical distributions $\mathrm{Cat}(K, \boldsymbol{\alpha})$ and $\mathrm{Cat}(K, \boldsymbol{\beta})$, which is evaluated by:

$$D_{\mathrm{KL}}(\boldsymbol{\alpha} \| \boldsymbol{\beta}) := \sum_{i=1}^{K} \alpha_i \log \frac{\alpha_i}{\beta_i}. \tag{3.12}$$

Adapt to model misspecification: A Neural Network classifier f trained on the source domain usually has poor performance when the source domain has a highly imbalanced label distribution ($d_{SU} \gg 0$) (Cao et al., 2019). In this case, the classifier is more likely to be subject to model misspecification error when estimating label shift. Hence we increase prior contribution in Eq. (3.10) with higher d_{SU}.

Adapt to sampling error: We propose two approaches to mitigate the problem that sampling error tends to increase given large label shift. Firstly, we use d_{TS} to approximate the amount of shift between target and source label distribution. A higher d_{TS} implies larger label shift, which will lead to more severe sampling error. Thus our APL model should increase the contribution of prior in Eq. (3.10) with higher d_{TS}. Secondly, when $\boldsymbol{\pi}^{MLE}$ is close to a uniform label distribution $\mathbf{1}/K$, we also propose to increase the prior contribution so that Eq. (3.10) can push the estimate more towards $\mathbf{1}/K$.

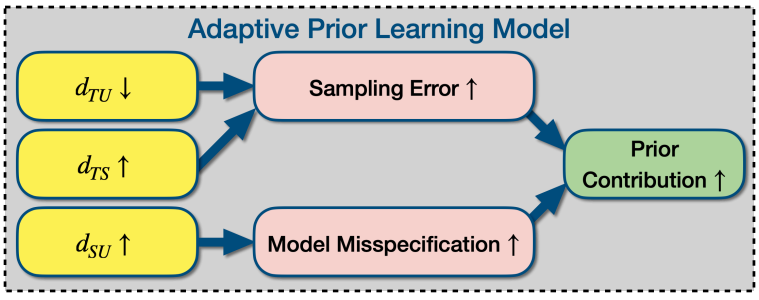

Figure 3.5: Structure of our Adaptive Prior Learning model. The parameter in the prior is adaptively determined by the available data with a heuristic based on d_{TU}, d_{TS}, d_{SU} defined in Eq. (3.11).

Overall APL model: By defining a normalization function $F(x) = x/(1+x)$, our APL model determines λ via:

$$\lambda = a \cdot F(\gamma \cdot d_{TU}) + (1-a) \cdot (1 - F(\gamma \cdot d_{TS})), \tag{3.13}$$

where $\gamma = 1 - F(b \cdot d_{SU})$ takes into account the model misspecification error and d_{TU}, d_{TS} evaluates the sampling error. Here $a \in [0,1]$ represents the trade-off between the two approaches to reduce sampling error and $b \in [0,1]$ represents the strength of misspecification error. More discussion of our APL model can be found in Section 3.4.3.

3.3.5 Sampling from the Bayesian posterior

Apart from the point estimate $\boldsymbol{\pi}^*$, we also use Bayesian analysis to utilize the entire posterior $p(\boldsymbol{\pi}|\mathcal{D}^t, \boldsymbol{\alpha})$ as our estimated target label distribution. In this Chapter, we propose to use the Markov Chain Monte Carlo (MCMC) method to obtain i.i.d. samples of the posterior. The samples can then be used for downstream label shift correction tasks.

Based on Eq. (3.4), we can rewrite $p(\boldsymbol{\pi}|\mathcal{D}^t, \boldsymbol{\alpha})$ as:

$$p(\boldsymbol{\pi}|\mathcal{D}^t, \boldsymbol{\alpha}) = \frac{1}{Z} p(\boldsymbol{\pi}|\boldsymbol{\alpha}) \prod_{i=1}^{N^t} \sum_{j=1}^{K} \frac{\pi_j}{c_j} f(x_i^t)_j, \tag{3.14}$$

where Z contains $\int p(\mathcal{D}^t|\boldsymbol{\pi}')p(\boldsymbol{\pi}'|\boldsymbol{\alpha}))d\boldsymbol{\pi}'$ and $p(X_t = x_i^t)$, which are constant w.r.t. $\boldsymbol{\pi}$ and are usually intractable.

To avoid evaluation of Z, we adopt the MCMC method to obtain samples of the posterior. Because the Hamiltonian Monte-Carlo (HMC) sampler can be more efficient than other MCMC methods in high dimensional space (Betancourt, 2017; Neal, 2011), we adopt the HMC to obtain i.i.d. samples of the posterior:

$$\Pi = \{\boldsymbol{\pi}^i\}_{i=1}^L, \text{ where } \boldsymbol{\pi}^i \sim_{i.i.d.} p(\boldsymbol{\pi}|\mathcal{D}^t, \boldsymbol{\alpha}), \tag{3.15}$$

where $\boldsymbol{\alpha}$ is determined by our APL model.

After collecting Π, each $\boldsymbol{\pi}^i$ is used as a point estimate of $\boldsymbol{\pi}$ for the downstream tasks. For example, for the label shift correction problem, we use every $\boldsymbol{\pi}^i \in \Pi$ to correct the source domain classifier $f(x)$ to the target domain $g_i(x)$ under label shift based on Eq. (2.34). The target domain average SoftMax classifier can then be constructed as:

$$g(x)_j = \sum_{i=1}^{L} \frac{1}{L} \frac{\frac{\pi_j^i}{c_j} f(x)_j}{\sum_{l=1}^{K} \frac{\pi_l^i}{c_l} f(x)_l}. \tag{3.16}$$

With samples of the posterior, the uncertainty of our estimated $\boldsymbol{\pi}$ given data can also be analyzed. Comparing with Algorithm 1, this approach utilizes the entire posterior at the cost of computation resources.

Remark: MCMC can be computationally expensive in high dimensional space, because sufficient warm up steps are required if the Markov chain is initialized randomly in value space (Mangoubi and Smith, 2017). Fortunately, since the posterior in our model is strictly log concave (Proposition 1) with known maximal point $\boldsymbol{\pi}^*$ obtained by MAPLS, we can initialize the Markov chain at $\boldsymbol{\pi}^*$ and the HMC sampler can then collect i.i.d. samples more efficiently without warm up steps.

3.3.6 Estimation of Source Label Distribution

Given source domain data $\{x_i^s, y_i^s\}_{i=1}^{N^s}$ and blackbox classifier f, there are two known methods to estimate the source domain label distribution $p_s(y = \cdot) = \boldsymbol{c}$. MLE is the standard method to estimate $\boldsymbol{c}$ with source domain ground truth labels y_i^s. On the other hand, when classifier f is calibrated on the source domain, Alexandari et al. (2020) also proposed to estimate $\boldsymbol{c}$ with source domain images in $\{x_i^s, y_i^s\}_{i=1}^{N^s}$ and classifier f with

$$c_j = \frac{1}{N^s} \sum_{i=1}^{N^s} f(x_i^s)_j. \tag{3.17}$$

In this Chapter, we adopt both approaches to estimate $\boldsymbol{c}$. We name the MLE approach as the “hard" method and Eq. (3.17) as the “soft" method.

3.3.7 Overall Framework

We propose to estimate and correct label shift as follows:

Algorithm 2 Overall Method

Input: Source domain data $\{x_i^s, y_i^s\}_i^{N^s}$, classifier $f : \mathcal{X} \to \Delta^{K-1}$ and target domain data $\mathcal{D}^t = \{x_i^t | (x_i^t, \cdot) \sim p_t(x, y), i = 1, 2, ...N^t\}$.
Parameter Determination:
- $\boldsymbol{c}$**:** Use MLE or Eq. (3.17) to estimate $P(Y_s = \cdot) = \boldsymbol{c}$.
- $\boldsymbol{\pi}^{MLE}$**:** Use MAPLS 1 to obtain $\boldsymbol{\pi}^{MLE}$ ($\alpha_0 = 1$).
- α_0**:** Determine λ with APL model.

With α_0 and Eq. (3.10), use MAPLS 1 to obtain $\boldsymbol{\pi}^*$.
if Point Estimate **then**
  Correct Label Shift: Obtain g with Eq. (2.34).
else if Posterior Sampling **then**
  Use HMC (initialized with $\boldsymbol{\pi}^*$) to obtain Π in Eq. (3.15).
  Correct Label Shift: Obtain g with Eq. (3.16).
end if
Output: Target domain classifier g.

We name the model that uses the MAP estimate MAPLS-APL and the model that uses posterior sampling PSLS-APL, where the “APL" indicates that parameter $\boldsymbol{\alpha}$ in the prior distribution is learned with our APL model.

3.4 Experiments

3.4.1 Experimental Setup

Datasets We evaluate our model on the CIFAR100 (Krizhevsky, Hinton et al., 2009), ImageNet 2012 (Russakovsky et al., 2015b) and Places2 (Zhou, Lapedriza et al., 2017) datasets. Following common use in Long-Tailed research (Cao et al., 2019; Wang, Lian et al., 2020; Zhong et al., 2021), we also use Long-Tail versions of ImageNet, Places (Liu, Miao, Zhan, Wang, Gong and Yu, 2019) and CIFAR100.

We test the models on test sets with Dirichlet shift proposed by previous label shift estimation models (Alexandari et al., 2020; Lipton et al., 2018). Dirichlet Shift generates a random test set label distribution from a K dimensional Dirichlet distribution. We also adopt the ordered Long-Tailed shifted test set used in LADE (Hong, Han et al., 2021), which has the same or inverse order of the Long-Tailed distributed train set. We further extend this setting to a shuffled Long-Tailed test set, where the test set still has a Long-Tailed label distribution but with random class order.

	Dataset	**Setup**
Train Set	CIFAR100 (Krizhevsky, Hinton et al., 2009)	Original, Long-Tailed with $R = \{2, 5, 10, 20, 50, 100, 200\}$
	ImageNet (Russakovsky et al., 2015b)	Original, Long-Tailed
	Places (Zhou, Lapedriza et al., 2017)	Original, Long-Tailed
Test Set	**Test Shift Type**	**Params**
	Original	None
	Dirichlet (Lipton et al., 2018)	$\alpha = 1.0, 10$
	Ordered Long-Tail (Hong, Han et al., 2021)	$R = \{2, 5, 10, 50\}$ Order = "Forward", "Backward"
	Shuffled Long-Tail	$R = \{2, 5, 10, 50\}$

Table 3.1: Closed SEt Label shift experiment settings. R is referred to as the imbalance ratio — the ratio of maximum and minimum sample number per class respectively in test set. α is the parameter of the Dirichlet distribution.

Model Setup Both our MAPLS/MAPLS-APL algorithm and previous MLLS algorithm are initialized with $\boldsymbol{\pi}^{(0)} = \boldsymbol{c}$ and run for 100 epochs to ensure convergence. Because $\boldsymbol{\pi}^*$ is unique as proved in Proposition 1, our MAPLS is guaranteed to converge to a single MAP estimate. In our APL model, we empirically set $a = 0.9, b = 0.5$ in Eq. (3.13) for all the label shift settings in all datasets. For our PSLS model, use a HMC sampler called No-U-Turn Sampler (Hoffman, Gelman et al., 2014) provided by Pyro (Bingham et al., 2018) to collect 5000 samples from the posterior.

We implement the Neural Network classifiers using PyTorch (Paszke et al., 2017). We use the ResNet32 (Idelbayev, n.d.) classifier for CIFAR100 and every CIFAR100-LT dataset. We use pre-trained ResNet50 (He, Zhang et al., 2016) and pre-trained Resnet152 for ImageNet and Places datasets respectively. We train a ResNet50 and

ResNet152 for ImageNet-LT and Places-LT datasets, respectively. More details of classifier implementations can be found in Appendix A.2.1.

Train Set \ Test Set	Ordered LT	Shuffled LT	Dirichlet
CIFAR100/CIFAR100-LT	55%	50%	52%
ImageNet/ImageNet-LT	82%	90%	75%
Places/Places-LT	68%	90%	92%

Table 3.2: SOTA comparison summary of estimation error. For all the label shift settings in Tab. 3.1, the percentage of settings that our MAPLS-APL model outperforms SOTA models (MLLS, BBSE, RLLS) in terms of $(w - \hat{w})^2/K$.

Train Set \ Test Set	Ordered LT	Shuffled LT	Dirichlet
CIFAR100/CIFAR100-LT	56%	58%	58%
ImageNet/ImageNet-LT	59%	80%	58%
Places/Places-LT	59%	80%	75%

Table 3.3: SOTA comparison summary of Top1 Accuracy. For all the label shift settings in Tab. 3.1, the percentage of settings that our MAPLS-APL model outperforms SOTA models and the baseline classifier in terms of accuracy.

Evaluation Metrics We follow previous methods (Alexandari et al., 2020; Lipton et al., 2018) to evaluate label shift estimation performance with $(w - \hat{w})^2/K$, where $w_i = p_t(y = i)/p_s(y = i), i = 1, 2...K$ is the target over the source label distribution ratio. w is the ground truth ratio estimated by the source and target labels. $\hat{w}$ is the predicted ratio with $p_t(y)$ estimated by each model.

We also provide Top1 accuracy for different label shift estimation models with LSC (Eq. (2.34)) on all datasets. The result summary is available in Tab. 3.3.

3.4.2 State-of-the-art Comparison

We compare the performance of our method with several state-of-the-art (SOTA) label shift estimation methods, including MLLS (Alexandari et al., 2020; Saerens et al., 2002), BBSE (Lipton et al., 2018) and RLLS (Azizzadenesheli et al., 2018). BBSE and RLLS also have “soft" and “hard" versions of each model. We evaluate the performance of these models with previously available implementation (details in Appendix A.2.2).

In large-scale datasets, methods that require retraining the classifier on the source domain will suffer from high computational costs. Therefore, we have not reproduced and reported Tachet des Combes et al. (2020) in our results.

We provide the SOTA comparison of our MAPLS-APL model in terms of $(w - \hat{w})^2/K$ in Tab. 3.2 and Top1 Accuracy in Tab. 3.3. Note that unlike SOTA models that obtain a point estimate of π, our PSLS-APL model obtains samples of π from the

Dataset	ImageNet								
Shift Type	Shuffled LT					Dirichlet			
Params	50	25	10	5	2	$\alpha = 10.0$		$\alpha = 1.0$	
Test sample No.	fixed	fixed	fixed	fixed	fixed	12500	25000	12500	25000
MLLS-hard	0.1210	0.1102	0.1001	0.0868	0.0766	0.1111	0.0848	0.1299	0.1113
MLLS-soft	**0.1121**	0.0972	0.0868	0.0721	0.0637	0.0981	0.0721	**0.1154**	**0.0977**
BBSE-hard	0.1285	0.1020	0.0871	0.0699	0.0581	0.0869	0.0661	0.1285	0.1173
BBSE-soft	0.1305	0.1086	0.0969	0.0790	0.0671	0.1052	0.0769	0.1366	0.1177
RLLS-hard	1.1450	0.7160	0.4436	0.2244	0.0473	0.1159	0.1122	1.1020	1.0607
RLLS-soft	1.1450	0.7160	0.4436	0.2244	0.0473	0.1159	0.1122	1.1020	1.0607
MAPLS-APL-hard (**Ours**)	0.1236	0.1006	0.0816	0.0633	0.0482	0.0736	0.0570	0.1283	0.1142
MAPLS-APL-soft (**Ours**)	0.1144	**0.0904**	**0.0710**	**0.0521**	**0.0370**	**0.0628**	**0.0465**	0.1160	0.1025

Table 3.4: Performance of $(w - \hat{w})^2/K$ (↓) on the ImageNet dataset, with shuffled Long-Tailed test set that have an imbalance ratio $\{50, 10, 5, 2\}$ and Dirichlet test set that have $\alpha = \{1, 10\}$ and total test sample number $\{12500, 25000\}$ in each setting. The best performances are in boldface, and the second best are in blue. Our PSLS-APL model is only suitable for Top1 Accuracy comparison.

Dataset	ImageNet-LT								
Shift Type	Shuffled LT					Dirichlet			
Params	50	25	10	5	2	$\alpha = 10.0$		$\alpha = 1.0$	
Test sample No.	fixed	fixed	fixed	fixed	fixed	12500	25000	12500	25000
MLLS-hard	36.09	34.49	30.57	26.90	24.42	28.44	26.03	38.21	36.18
MLLS-soft	80.66	82.10	84.92	81.54	76.59	91.28	83.62	82.62	84.23
BBSE-hard	$3.2e^5$	$1.8e^6$	$1.4e^6$	$2.0e^7$	$4.8e^5$	$4.8e^5$	$1.0e^7$	$1.7e^6$	$1.2e^{10}$
BBSE-soft	28.00	25.48	18.04	15.86	12.07	13.84	12.89	28.30	27.75
RLLS-hard	45.00	38.77	29.99	24.18	19.96	21.98	21.05	46.05	45.75
RLLS-soft	45.00	38.77	29.98	24.18	19.96	21.98	21.05	46.05	45.75
MAPLS-APL-hard (**Ours**)	20.25	16.62	**10.26**	**6.18**	**2.62**	**4.72**	**3.86**	21.16	20.68
MAPLS-APL-soft (**Ours**)	**19.48**	**16.39**	11.23	7.58	4.43	6.62	5.66	**18.94**	**18.75**

Table 3.5: Performance of $(w - \hat{w})^2/K$ (↓) on the ImageNet-LT dataset, with shuffled Long-Tailed test set that have an imbalance ratio $\{50, 10, 5, 2\}$ and Dirichlet test set that have $\alpha = \{1, 10\}$ and total test sample number $\{12500, 25000\}$ in each setting. The best performances are in boldface, and the second best are in blue. Our PSLS-APL model is only suitable for Top1 Accuracy comparison.

posterior instead. Thus only Top1 Accuracy is compared for our PSLS-APL model (Tab. 3.6) instead of both metrics.

As shown in Tab. 3.2, our MAPLS-APL model outperforms SOTA models in at least 50% of the label shift and dataset settings. As an example on ImageNet in Tab. 3.4, our model outperforms other models by a large margin for the highly imbalanced train set ImageNet-LT.

As shown in Tab. 3.3, in terms of Top1 Accuracy, our MAPLS-APL model outperforms SOTA models and baseline in at least 50% of the settings. As an example in Tab. 3.6, our MAPLS-APL and PSLS-APL model have similar performance and outperform SOTA models in most settings.

By analyzing the performance in Tab. 3.4, 3.5, 3.6, 3.7, one obvious advantage of our model is its robustness to the source label distribution. When the source domain has a highly imbalanced label distribution (*e.g.* ImageNet-LT, Places-LT), the label shift

Dataset	ImageNet-LT								
Order	Forward				Uniform	Backward			
Imbalance Ratio	25	10	5	2	1	2	5	10	25
Baseline	**62.55**	**58.48**	54.97	49.59	45.31	40.94	35.22	31.31	26.56
MLLS-hard	59.10	55.42	52.70	48.93	46.47	44.04	41.27	39.82	38.30
MLLS-soft	58.45	54.70	52.13	48.56	46.34	44.11	41.66	40.41	39.30
BBSE-hard	33.20	25.93	24.53	19.03	26.15	23.99	16.85	28.03	15.67
BBSE-soft	60.95	57.47	54.86	51.03	48.23	45.67	42.47	40.42	37.94
RLLS-hard	**62.55**	**58.48**	54.97	49.59	45.31	40.94	35.22	31.31	26.56
RLLS-soft	**62.55**	**58.48**	54.97	49.59	45.31	40.94	35.22	31.31	26.56
MAPLS-APL-hard (**ours**)	60.67	57.72	**55.56**	52.51	50.31	48.05	45.09	43.33	41.31
MAPLS-APL-soft (**ours**)	60.34	57.58	55.44	52.50	50.32	**48.33**	**45.69**	**44.21**	**42.50**
PSLS-APL-hard (**ours**)	60.80	58.00	55.49	52.65	50.34	47.88	45.07	43.27	41.33
PSLS-APL-soft (**ours**)	60.61	57.95	55.46	**52.76**	**50.45**	48.00	45.47	43.86	42.22

Table 3.6: Performance of Top1 Accuracy (↑) on ImageNet-LT dataset, with Ordered Long-Tailed test set that have imbalance ratio $R = \{25, 10, 5, 2\}$. The best performances are in boldface, and the second best are in blue.

Dataset	Places-LT								
Order	Forward				Uniform	Backward			
Imbalance Ratio	25	10	5	2	1	2	5	10	25
Baseline	41.25	38.04	35.11	31.06	27.92	24.76	20.89	18.26	15.49
MLLS-hard	40.46	37.78	35.67	32.95	31.08	29.22	26.85	25.30	23.70
MLLS-soft	39.90	37.20	35.06	32.43	30.53	28.72	26.58	25.50	24.11
BBSE-hard	28.65	28.39	27.83	26.37	26.79	24.51	23.09	16.69	17.60
BBSE-soft	41.12	38.32	36.18	33.16	30.94	28.75	26.16	24.34	22.31
RLLS-hard	41.25	38.04	35.11	31.06	27.92	24.76	20.89	18.26	15.72
RLLS-soft	41.25	38.04	35.11	31.06	27.92	24.76	20.89	18.26	15.49
MAPLS-APL-hard (**ours**)	41.34	**39.55**	38.01	**36.04**	34.48	32.80	30.49	28.68	26.63
MAPLS-APL-soft (**ours**)	41.15	39.32	37.78	36.04	**34.58**	32.97	**30.87**	**29.35**	**27.41**
PSLS-APL-hard (**ours**)	**41.61**	39.19	**38.14**	36.01	34.49	**32.99**	30.44	28.64	26.74
PSLS-APL-soft (**ours**)	41.44	39.11	38.11	36.01	34.52	32.97	30.51	28.89	27.17

Table 3.7: Performance of Top1 Accuracy (↑) on Place-LT dataset, with Ordered Long-Tailed test set that have imbalance ratio $R = \{25, 10, 5, 2\}$. The best performances are in boldface, and the second best are in blue.

estimation performance of our model stays relatively stable while previous models degrade significantly.

3.4.3 Ablation Study

Posterior Sampling Results: We provide the density histogram of 5000 posterior samples Π collected by our PSLS-APL model in Fig. 3.6, with single value of $\boldsymbol{\pi}$ estimated by other models as well. The posterior $p(\boldsymbol{\pi}|\mathbb{X}, \boldsymbol{\alpha})$ fits well with the ground truth and is able to provide a sense of uncertainty of our estimation.

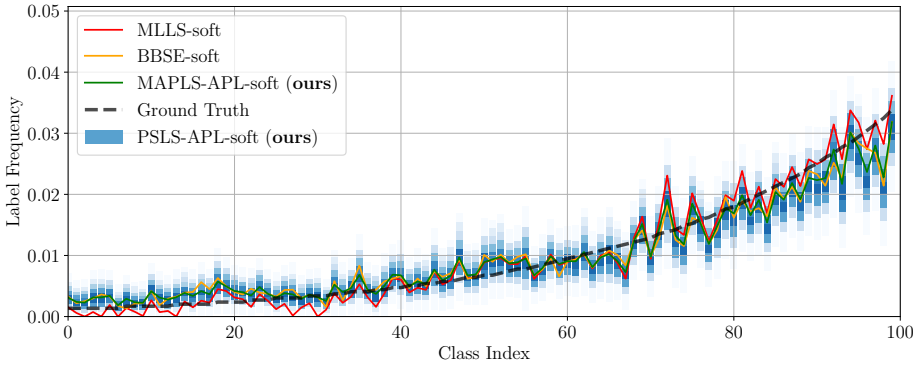

Figure 3.6: Illustration of the label shift estimation result (π). On the Long-Tailed CIFAR100 dataset with the Ordered Long-Tailed test set, our PSLS-APL model uses HMC to obtain 5000 i.i.d. samples of the posterior $p(\boldsymbol{\pi}|\boldsymbol{\alpha}, \mathbb{X})$ (posterior sample density histogram plot as blue bar heatmap), which fit nicely with the ground truth.

Estimation Error During Classifier Training: We also analyze the estimation stability of our model during the training of classifier f on the source domain. Specifically, we monitor the performance of each label shift estimation model during the training of a Neural Network classifier on the Long-Tailed CIFAR100 dataset. The test sets have Ordered Long-Tailed label distribution. As shown in Fig. 3.7, the performance of BBSE, MLLS and our model improves during the training of the classifier. This observation suggests that the label shift estimation performance of these models could be further enhanced with a better classifier. Our MAPLS ($\lambda = 0.9$) and MAPLS-APL model performs better and stable in the last 50 epochs.

Empirical Justification of APL model: With empirical evidence, we show that the best choice of the parameter for the Dirichlet prior $\boldsymbol{\alpha}$ or parameter $\lambda = N/(N + K(\alpha_0 - 1))$ in our MAPLS model is different with different source and target label distribution settings. Further, our proposed Adaptive Prior Learning model can give a good choice of λ.

We compared the performance of the MAPLS-APL model with MAPLS models that have fixed λ. The experiment is carried out on the ImageNet-LT dataset with a uniform test set. As shown in Eq. (3.8), our MAPLS model degenerates to MLLS in the $\lambda = 1$ setting. As seen from Tab. 3.8, the best choice of λ varies with different target label distributions. MLLS generally performs worse than every MAPLS model. Our heuristic can give the λ that is close to the best choice.

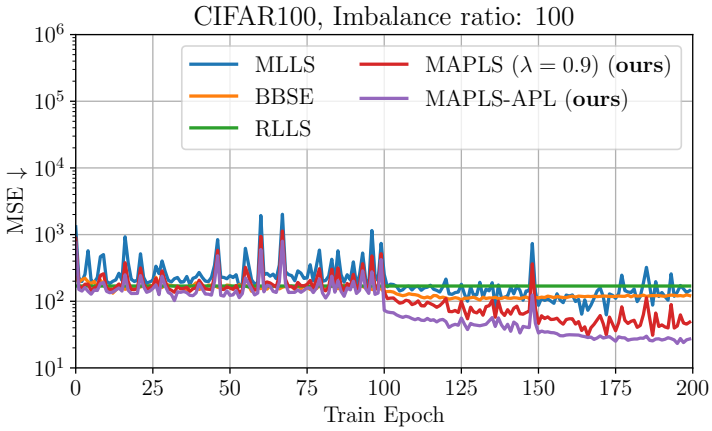

Figure 3.7: Ablation study on stability of the MAPLS-APL model. On the Long-Tailed CIFAR100 dataset with Ordered LT test set, our model is stable during the training of the classifier and performs better than SOTA methods.

Order	Forward					Uniform	Backward				
Imbalance Ratio	50	25	10	5	2	1	2	5	10	25	50
MLLS-soft	48.66	42.51	40.37	58.19	73.82	83.77	86.10	161.63	163.43	252.68	291.19
MAPLS-soft ($\lambda = 0.9$)	**12.92**	**12.30**	13.23	15.09	14.66	18.66	20.29	45.51	52.17	98.66	133.56
MAPLS-soft ($\lambda = 0.7$)	15.88	14.62	**12.7211**	**10.34**	6.19	4.68	4.41	12.41	22.51	**51.81**	**82.34**
MAPLS-soft ($\lambda = 0.5$)	20.83	18.76	15.27	11.36	**5.52**	2.48	1.78	8.89	**21.69**	52.75	86.87
MAPLS-soft ($\lambda = 0.3$)	24.94	22.20	17.55	12.62	5.63	1.90	1.17	**9.19**	24.33	59.67	98.37
MAPLS-soft ($\lambda = 0.1$)	28.15	24.90	19.40	13.74	5.90	**1.76**	**1.10**	10.21	27.28	66.61	109.39
MAPLS-APL-soft (**ours**)	13.82	13.38	12.55	10.39	5.80	3.59	3.67	13.48	24.40	57.78	90.37

Table 3.8: Performance of $(w - \hat{w})^2/K$ on ImageNet-LT dataset, with Ordered Long-Tailed test sets that have imbalance ratio $R = \{50, 10, 5, 2\}$ and forward and backward order. Best among fixed λ models are in boldface. Each reported value is the average of 10 in different shuffled and random sampled test sets.

EM algorithm convergence analysis: For MLLS and our proposed model, we follow MLLS (Alexandari et al., 2020) to initialize target label distribution the same as source domain label distribution $\boldsymbol{\pi}^{(0)} = \boldsymbol{c}$. Each EM have $T = 100$ iteration to get the final estimation. As shown in Fig. 3.8, MLLS and our MAPLS-APL algorithm have converged after 100 iteration.

Robustness Study We also study the robustness of our MAPLS model when Assumption 3 is not satisfied. As discussed by previous works (Garg, Wu, Balakrishnan et al., 2020), when Assumption 3 holds, it implies that classifier f is perfectly calibrated on the source domain. Since the calibration performance of the classifier is usually evaluated through Expected Calibration Error (ECE) (Guo, Pleiss et al., 2017) we plot the estimation error and ECE correlation of our model in Fig. 3.9 and estimation error and Top1 Accuracy correlation of our model in Fig. 3.10. The data points in these two figures are collected during the training phase of a ResNet32 classifier on the CIFAR100 dataset, with the model tested on an Ordered Long-Tailed test set ("Forward" LT100).

As shown in the figures, the estimation error $(w - \hat{w})^2/K$ of both our model and the

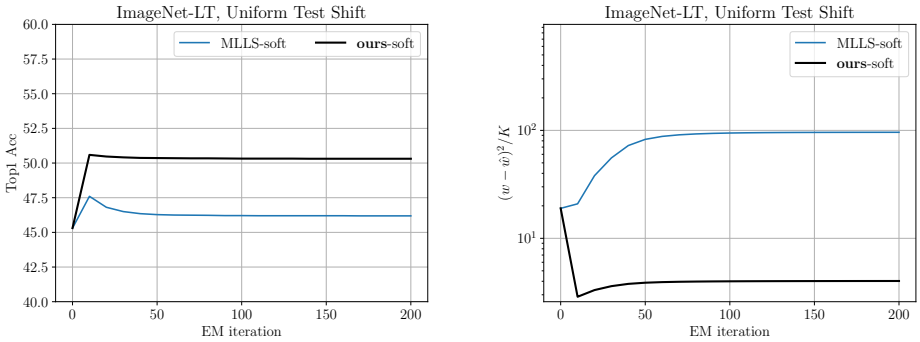

Figure 3.8: EM algorithm convergence analysis. Performance of EM algorithm MLLS-soft and our MAPLS-APL-soft with different number of EM iterations, on ImageNet-LT dataset with uniform test set. Each algorithm have converged after 100 iteration.

MLLS model are not sensitive to the calibration performance ECE of the classifier. On the other hand, it seems the estimation performance increases when the classifier exhibits higher accuracy. Both of these observations imply that, our model is robust to both the calibration error ECE and accuracy of an image classifier.

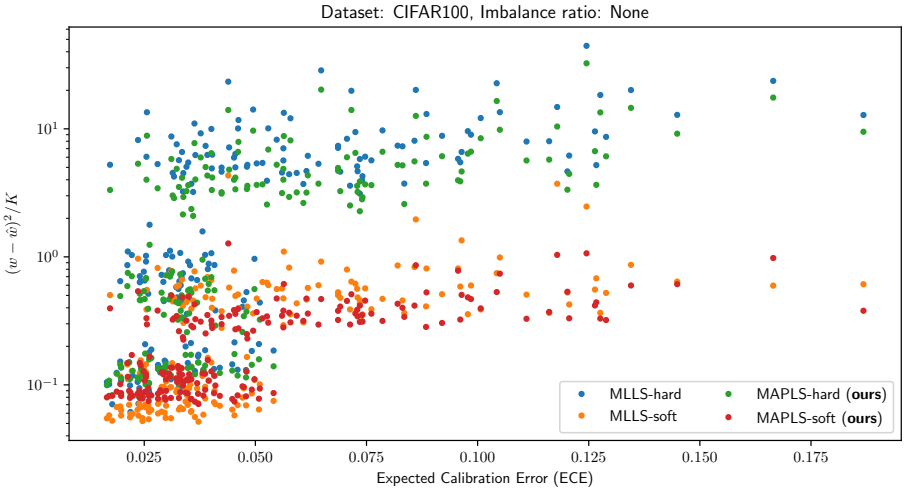

Figure 3.9: Estimation error and classifier calibration performance (ECE) correlation of our MAPLS model and previous MLLS model. The samples are collected when training a classifier on the CIFAR100 dataset. The estimation error $(w-\hat{w})^2/K$ of our model and the MLLS model are not sensitive, *i.e.*robust, w.r.t. the calibration performance ECE of the classifier.

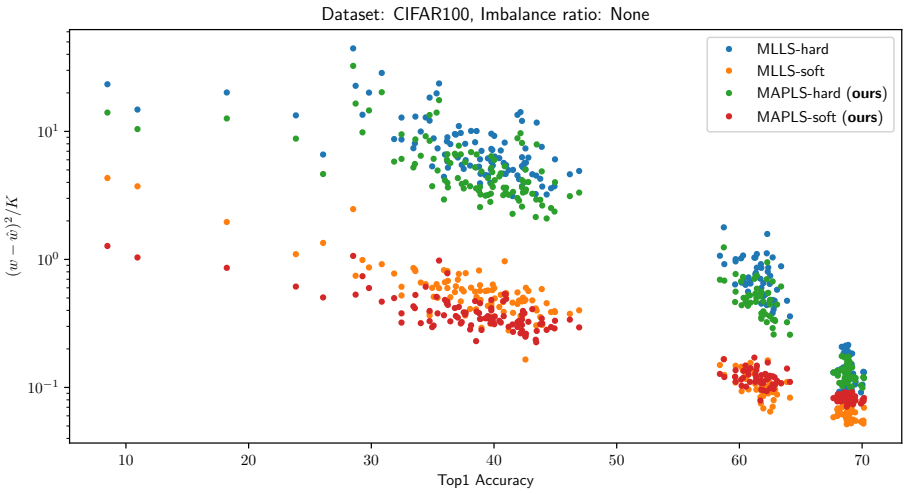

Figure 3.10: Estimation error and Top1 Accuracy correlation of our MAPLS model and previous MLLS model. The samples are collected when training a classifier on CIFAR100 dataset. The estimation error $(w-\hat{w})^2/K$ of our model and the MLLS model are not sensitive, *i.e.*robust, to the classification Top1 Accuracy of the classifier.

3.5 Conclusion

In this chapter, we developed label shift estimation methods MAPLS-APL and PSLS-APL under a Bayesian framework that are applicable to large-scale datasets and robust to highly imbalanced source label distributions. In our MAPLS model, we derived an EM algorithm to obtain the MAP estimate of the target label distribution and proposed a novel Adaptive Prior Learning model to adaptively adjust the prior parameter. In our PSLS model, we used HMC to sample from the strictly log-concave posterior $p_t(\boldsymbol{\pi}|\mathcal{D}^t, \boldsymbol{\alpha})$.

Unlike previous benchmark evaluations, our experimental settings additionally covers a variety of large-scale datasets (ImageNet, Places) with highly imbalanced label distributions, which provide a more realistic evaluation of SOTA methods. Experiments on these datasets have demonstrated the effectiveness of our model and its potential to be applied in real world label shift problems.

Chapter 4

Feature Based Closed Set Label Shift

4.1 Introduction

Similar to Chapter 3, this chapter also focuses on the Closed Set Label Shift *estimation* problem. We aim to develop a CSLS estimation model that is feasible for a classifier that outputs image features instead of probability simplexes discussed in Chapter 3. The model proposed in this chapter can be seen as an extension of the model proposed in Chapter 3.

Refresher on the CSLS problem In the Closed Set Label Shift (OSLS) problems, we are given a source domain labeled dataset, a target domain unlabeled dataset and a source domain classifier. The objectives include 1) *detection*: detect if the label distributions between the source and target domain are identical, 2) *estimation*: estimate the target label distribution and 3) *correction*: construct an appropriate target domain classifier. The mathematical definition is given in the next section (duplicating the original definition in Chapter 2).

Objectives We have two objectives for our proposed model. Firstly, we aim at developing a general label shift estimation model for different types of Neural Network (NN) classifiers, including SoftMax classifiers or prototype classifiers (Bezdek and Kuncheva, 2001; Snell et al., 2017a). Such a model could extend the application regime of existing works, which are usually designed for some specific types of classifiers. For example, BBSE (Lipton et al., 2018), RLLS (Azizzadenesheli et al., 2018) are based on discrete label classifiers, while MLLS (Alexandari et al., 2020) requires a calibrated classifier (Garg, Wu, Balakrishnan et al., 2020). 2) Similar to Chapter 3, we also want to ensure that the new model performs robustly in highly class-imbalanced large-scale datasets, which are commonly seen in real-world problems (Liu, Miao, Zhan, Wang, Gong and Yu, 2019).

Contributions In this chapter, we propose a Generalized Label Shift Estimation (GLSE) model to achieve the two objectives. Our GLSE model provides a general framework for constructing practical label shift estimation algorithms using different types of classifiers. The framework has two stages: a Class Conditional Model (CCM) and a Latent Variable Model (LVM). Under the GLSE framework, we derive EM algorithms to obtain Maximum Likelihood estimates (MLE) and Maximum *a Posteriori* (MAP) estimates of target label distributions. We also use Markov Chain

Monte Carlo to sample the target label distribution from the Bayesian posterior. We further propose a novel Adaptive Prior Learning (APL) model to determine the parameter in the Bayesian prior with data.

We conduct extensive experiments on large-scale datasets with different label distributions. We test our model on CIFAR100, ImageNet, Places and Long-Tailed versions of each dataset. Experimental results show that our model consistently outperforms existing state-of-the-art (SOTA) models, particularly, obtaining better performance when the source or target label distribution is highly imbalanced. These results demonstrate the robustness of our model to different types of label shift and its applicability to real world classification problems.

We summarise the contributions of this Chapter:

1. We propose a Generalized Label Shift Estimation (GLSE) model. Our model provides a framework that enables the generation of different practical label shift estimation algorithms with different choices of model setup, including SOTA model MLLS.
2. In our GLSE framework, we construct the MLE and MAP objectives of the target label distribution given data, prove the convexity of the objectives and derive EM algorithms that converge to the MLE/MAP estimate of the target label distribution.
3. We adopt the MCMC method to obtain i.i.d. samples from the posterior of the target label distribution. We propose an Adaptive Prior Learning (APL) model to determine the parameters of the Bayesian prior given data.
4. Experiments show that the models constructed from our GLSE framework outperform previous SOTA models on various label shift settings on CIFAR100, ImageNet, Places and Long-Tailed version of each dataset.

Background and Related Works The background about the Latent Variable Model (LVM) and EM algorithm used in this chapter can be found in Chapter 2, Section 2.2.4 and 2.2.5, respectively. The related literature on the CSLS problem can be found in Chapter 2, Section 2.3.1.

4.2 Problem Setup and Analysis

We focus on the label shift problem in the closed set image classification tasks, where we denote the image space as $\mathcal{X} \subseteq \mathbb{R}^d$ and the label space as $\mathcal{Y} = \{1, 2, ..., K\}$. $\mathcal{Z}$ denotes the feature space and $f : \mathcal{X} \to \mathcal{Z}$ denotes the corresponding feature extractor. Let X, Y, Z be the random variables (RVs) of image, label and feature respectively.

4.2.1 Definition and Assumptions

In the label shift problem, we have a labeled source domain dataset $\mathcal{D}^s$ and an unlabeled target domain dataset $\mathcal{D}^t$ available, where $\mathcal{D}^s = \{(x_i^s, y_i^s)\}_{i=1}^{N^s}$ contains

image and label jointly drawn i.i.d. from $p_s(x,y)$ and $\mathcal{D}^t = \{x_i^t\}_{i=1}^{N^t}$ contains images drawn i.i.d. from the marginal $p_t(x)$.

In the closed set label shift (CSLS) problem, recall that we have the closed set label shift assumption available in Chapter 2, Assumption 1:

Assumption 1. (Closed Set Label Shift Assumption)

$$p_s(x|y=i) = p_t(x|y=i) \quad \text{for all} \quad i \in \mathcal{Y}. \tag{2.1}$$

Under Assumption 1, we are given with a labeled source domain dataset $\mathcal{D}^s$, an unlabeled target domain data $\mathcal{D}^t$ and a source domain classifier f. The *detection*, *estimation* and *correction* problems are then defined as:

Definition 1. (Closed Set Label Shift Problem)

Under Assumption 1, given:

- Source domain labeled data $\mathcal{D}^s = \{(x_i^s, y_i^s)\}_{i=1}^{N^s}$ (K classes);
- Target domain unlabeled data $\mathcal{D}^t = \{x_i^t\}_{i=1}^{N^t}$ (K classes);
- Source domain classifier f.

the CSLS problem is to solve

- *Detection*: Verify $p_s(y=\cdot) = p_t(y=\cdot)$;
- *Estimation:* Estimate $p_t(y=\cdot)$;
- *Correction:* Model $p_t(y=\cdot|x)$ based on f.

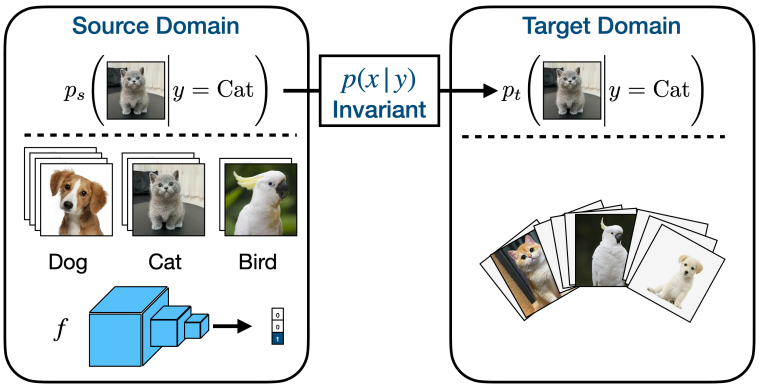

Figure 4.1: Closed Set Label Shift problem setup, where we are given: 1) **assumption**: the closed set label shift assumption (Assumption 1), 2) **datasets**: a source domain labeled dataset and a target domain unlabeled dataset and 3) **model**: a source domain classifier or feature extractor f. This information is used for the closed label shift ***detection***, ***estimation*** and ***correction*** problems.

In this Chapter, we adopt a relaxed condition that a feature extractor $f : \mathcal{X} \to \mathcal{Z}$ is available. Different to Chapter 3 or previous works (Garg, Wu, Balakrishnan et al., 2020) that assume the availability of a SoftMax classifier or a blackbox classifier (Lipton et al., 2018), this setting allows $\mathcal{Z}$ to be the same as the hard label space

$\mathcal{Y}$, soft label space (probability simplex space) Δ^{K-1} or feature space of a prototype classifier, thereby generalising previous settings. The final classifier $f_0 : \mathcal{X} \to \mathcal{Y}$ can be constructed based on the feature extractor f.

In Definition 1, the feature extractor f can be obtained by training a NN model on the source domain dataset $\mathcal{D}^s$ via supervised learning. Our main assumption is that for a certain choice of feature extractor f, the conditional distribution of feature z given label y belongs to some distribution family:

Assumption 4. (**CSLS Feature Based Approach Assumption**)

The conditional distribution of the features given the label belongs to a known family $\mathcal{Q}$:

$$p_s(z|y) \in \mathcal{Q} = \{q(\cdot|\cdot,\theta)|\theta \in \Theta\}. \tag{4.1}$$

Here $\mathcal{Q}$ is a space of probability density/mass functions that are supported on feature space $\mathcal{Z}$ for all labels in $\mathcal{Y}$. Θ is the corresponding parameter space. $q(z_0|i,\theta)$ denotes the evaluation of the function at $z_0 \in \mathcal{Z}$ given label $i \in \mathcal{Y}$ and parameter $\theta \in \Theta$. The distribution family $\mathcal{Q}$ can be constructed based on f and corresponding $\mathcal{Z}$ space, which is further discussed in Section 4.3.3.

We superimpose our assumptions onto the graphical model in Figure 4.2.

4.2.2 Graphical Model Setup

Observing that RVs of the source and target domain label Y_s, Y_t are supported on finite and discrete set $\mathcal{Y} = \{1, 2, ..., K\}$, Y_s, Y_t can be seen as following categorical distributions $\text{Cat}(K, \cdot)$ over K classes.

Here we denote $Y_s \sim \text{Cat}(K, \boldsymbol{c})$ and $Y_t \sim \text{Cat}(K, \boldsymbol{\pi})$. The parameters $\boldsymbol{c}, \boldsymbol{\pi}$ of the categorical distributions are K dimensional probability simplexes that satisfy $p_s(y = i) = c_i$ and $p_t(y = i) = \pi_i$ for all $i \in \mathcal{Y}$. Thus we have $\boldsymbol{c}, \boldsymbol{\pi} \in \Delta^{K-1}$ where Δ^{K-1} denotes the space of K dimensional probability simplex. Estimating the target label distribution $p_t(y)$ is equivalent to estimating parameters $\boldsymbol{\pi}$ in $\text{Cat}(K, \boldsymbol{\pi})$.

The graphical model setup along with our assumptions are illustrated in Fig. 4.2. We optionally place priors over the parameter $\boldsymbol{\pi}$ of the target label distribution $p_t(y = \cdot) = \boldsymbol{\pi}$, which is further discussed in subsequent sections. We treat the parameter $\boldsymbol{c}$ of the source label distribution $p_s(y = \cdot) = \boldsymbol{c}$ as deterministic parameters, which can be directly estimated with the source domain labeled dataset $\mathcal{D}^s$.

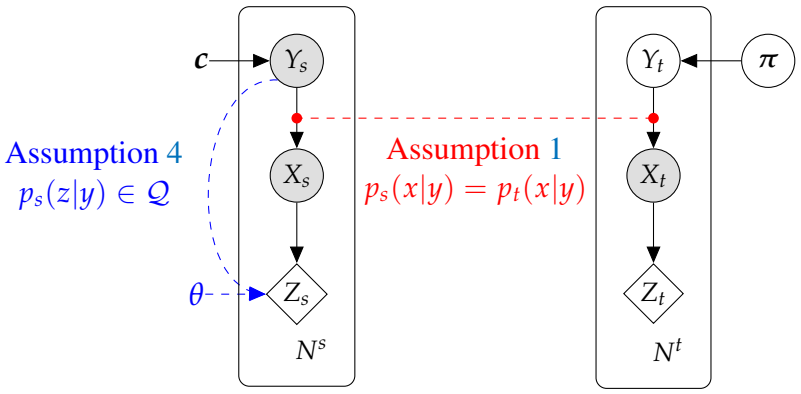

Figure 4.2: Graphical model of the feature based Closed Set Label Shift setting and our assumptions. X_s, X_t are data for the source and target domain, Y_s, Y_t are the corresponding categorical-valued labels and Z_s, Z_t are corresponding image features. $\boldsymbol{c}, \boldsymbol{\pi}$ are parameters of the source and target domain label distributions respectively. The source domain random variable of image X_s and label Y_s are observed with i.i.d. samples available in $\mathcal{D}^s$. The target domain random variable of image X_t is observed with i.i.d. samples available in $\mathcal{D}^t$. With the closed set label shift assumption (Assumption 1), we estimate $\boldsymbol{\pi}$ by further assuming $p_s(z|y=\cdot)$ belongs to some distribution family $\mathcal{Q}$ (Assumption 4).

4.3 Proposed Method

4.3.1 Model Overview

This section proposes a General Label Shift Estimation (GLSE) model for the Closed Set Label Shift (CSLS) *estimation* and *correction* tasks.

Method Summary: Our objective is to design a CSLS *estimation* model that is feasible when the image classifier outputs different forms of predictions, which include discrete labels, SoftMax probabilities and image features for a prototype-based classifier.

Roughly speaking, this objective is achieved by viewing the CSLS *estimation* problem as estimating the mixture weight of a distribution mixture (*e.g.* Gaussian Mixture), where each mixture component is the per-class distribution of the image features output from a classifier (*e.g.* Gaussian).

Formally speaking, we reformulate the CSLS *estimation* problem in the form of a Latent Variable Model, with the target label as the unobserved latent variable. In the LVM, the distribution family of $p_s(f(x)|y)$ is specified in **Stage 1** (Section 4.3.3) for different choice of classifier. The estimate of the target label distribution is obtained in **Stage 2** (Section 4.3.4) through deriving EM algorithms to estimate the parameter of the distribution of the latent variable – the target label in the LVM. A visualization of the model structure is provided in Fig. 4.3.

Content Outline:

1. (Section 4.3.2) The general idea of the proposed model is elaborated.

2. (Section 4.3.3) In **Stage 1**, the Class Conditional Model is proposed to determine distribution family $\mathcal{Q}$ and estimate the corresponding parameter θ.
3. (Section 4.3.4) In **Stage 2**, the Latent Variable Model is proposed to obtain a MLE or the MAP estimate of $p_t(y = \cdot) = \boldsymbol{\pi}$.
4. (Section 4.3.5) A practical example of our GLSE model is provided.

The overall feature based CSLS estimation and correction model is summarized in Section 4.3.6. All the theoretical proofs in our model are available in Appendix B.1.

4.3.2 The Two Stage Approach

Our GLSE model analyzes label shift in the feature space $\mathcal{Z}$. With feature extractor f, we pre-process the datasets $\mathcal{D}^s, \mathcal{D}^t$ to obtain the labeled source domain features $\mathcal{D}_z^s = \{(z_i^s, y_i^s) | z_i^s = f(x_i^s), (x_i^s, y_i^s) \in \mathcal{D}^s\}$ and unlabeled target domain features $\mathcal{D}_z^t = \{z_i^t | z_i^t = f(x_i^t), x_i^t \in \mathcal{D}^t\}$.

In feature space $\mathcal{Z}$, we also have:

Lemma 2. *(Lipton et al., 2018)*

Under Assumption 1, if $f : \mathcal{X} \to \mathcal{Z}$ is deterministic, then for $z = f(x)$ we have:

$$p_s(z|y = i) = p_t(z|y = i) \quad \textit{for all} \quad i \in \mathcal{Y}. \tag{4.2}$$

Based on the Closed Set Label Shift assumption (Assumption 1) and Lemma 2, the marginal distribution of the target domain feature $Z_t \sim p_t(z)$ can be decomposed w.r.t. the target domain label Y_t as:

$$\underbrace{p_t(z)}_{\substack{Z_t \text{ observed} \\ \text{samples in } \mathcal{D}_z^t}} = \sum_{i=1}^{K} \underbrace{p_s(z|y = i)}_{\substack{p_s(z|y) \in \mathcal{Q} \text{(Assumption 4)} \\ \text{samples in } \mathcal{D}_z^s}} \underbrace{p_t(y = i)}_{\substack{Y_t \text{ latent} \\ \text{(Objective)}}}. \tag{4.3}$$

Eq. (4.3) can be seen as a **Latent Variable Model** (LVM) discussed in the previous Chapter 2.2.4, where the target domain label Y_t is the unobserved latent variable and the target domain feature Z_t is the observed variable with available i.i.d. samples given in target unlabeled dataset $\mathcal{D}_z^t$. The conditional distribution $p_s(z|y = i)$ that connects the observed and latent variable has its distribution family specified based on Assumption 4. Moreover, i.i.d. samples of $p_s(z|y)$ are available in the source domain dataset $\mathcal{D}_z^s$.

Based on the above observation, a two stage label shift estimation model can be constructed. In **Stage 1**, we can specify the distribution family of $\mathcal{Q}$ that $p_s(z|y)$ belongs (*e.g.* Gaussian family of distributions). And the parameter θ of the $p_s(z|y) \in \mathcal{Q}$ is estimated based on the features in the source domain dataset $\mathcal{D}_z^s$ for each corresponding label $i \in \mathcal{Y}$. In **Stage 2**, the estimate the target domain label distribution $p_t(y = \cdot) = \boldsymbol{\pi}$ is estimated in the LVM (Eq. (4.3)), where $\boldsymbol{\pi}$ is the parameter of the distribution of the latent variable Y_t.

More specifically, with Assumption 1 and Assumption 4, we estimate target label distribution $p_t(y=\cdot)=\boldsymbol{\pi}$ via:

- **Stage 1:** Construct distribution family $\mathcal{Q}$ in Assumption 4 based on the choice of f. Obtain point estimate θ^* for parameter θ in $p_s(z|y)=q(\cdot|\cdot,\theta)$ with the source domain dataset $\mathcal{D}_z^s$.
- **Stage 2:** Construct a LVM by viewing Y_t as the latent variable in Eq. (4.3). Estimate target label distribution $p_t(y=\cdot)=\boldsymbol{\pi}$ with the target domain dataset $\mathcal{D}_z^t$ and $q(\cdot|\cdot,\theta^*)$ obtained in **Stage 1**.

We name **Stage 1** as the Class Conditional Model. The details are discussed in Section 4.3.3, where several examples of $f, \mathcal{Q}$ are given and their relationship with previous works are discussed. We discuss details of the LVM model in **Stage 2** in Section 4.3.4, where we derive EM algorithms that obtain an MLE/MAP estimate of the target label distribution $p_t(y=\cdot)=\boldsymbol{\pi}$.

Remark: Note that in **Stage 1**, both $\mathcal{D}_z^s, \mathcal{D}_z^t$ can be used to obtain a point estimate of θ' under label shift. We provide more discussion about this setting in Section B.1.6.

4.3.3 Stage 1: Class Conditional Model

The distribution family $\mathcal{Q}=\{q(\cdot|i,\theta)|\theta\in\Theta\}$ can be constructed based on the choice of f. In this section, we provide examples of f and corresponding $\mathcal{Q}$ and means to obtain a point estimate of θ that satisfies $p_s(z|y=i)=q(\cdot|\cdot,\theta)$ given the source domain labeled features in $\mathcal{D}_z^s$.

Choices of Distribution Family $\mathcal{Q}$:

Discrete $\mathcal{Z}$ Space: When the feature space is finite and discrete: $\mathcal{Z}=\{1,2,...,M\}$, the conditional distribution $p_s(z|y=j)$ for class $j\in\mathcal{Y}$ is categorical, thus we can define the $q(\cdot|j,\theta)$ and parameter space Θ in $\mathcal{Q}$ as:

$$\begin{cases} q(z|j,\theta)=\theta_z^j \\ \Theta=\left\{\left[\theta^1,\theta^2,\ldots,\theta^K\right]\middle|\theta^i\in\Delta^{M-1}\right\}. \end{cases} \tag{4.4}$$

Here Θ contains the parameters of K independent M dimensional categorical distributions. We use GLSE_{Cat} to denote the GLSE model with $\mathcal{Q}$ defined in Eq. (4.4).

Such a $\mathcal{Q}$ space can be seen as a space of $M\times K$ stochastic matrices, which generalizes the setting of BBSE (Lipton et al., 2018), where BBSE requires $M=K$, *i.e.*, feature space has the same dimension as the number of classes K in the dataset.

SoftMax $\mathcal{Z}$ Space (Case 1): When the feature space $\mathcal{Z}=\Delta^{M-1}$, f can be seen as a SoftMax classifier for M classes. In this case, $p_s(z|y=j)$ is supported on Δ^{M-1}, for which every distribution may be represented as a Dirichlet distribution. Therefore we can assume $p_s(z|y=j)$ belongs to the Dirichlet family of distributions. The $q(\cdot|j,\theta)$

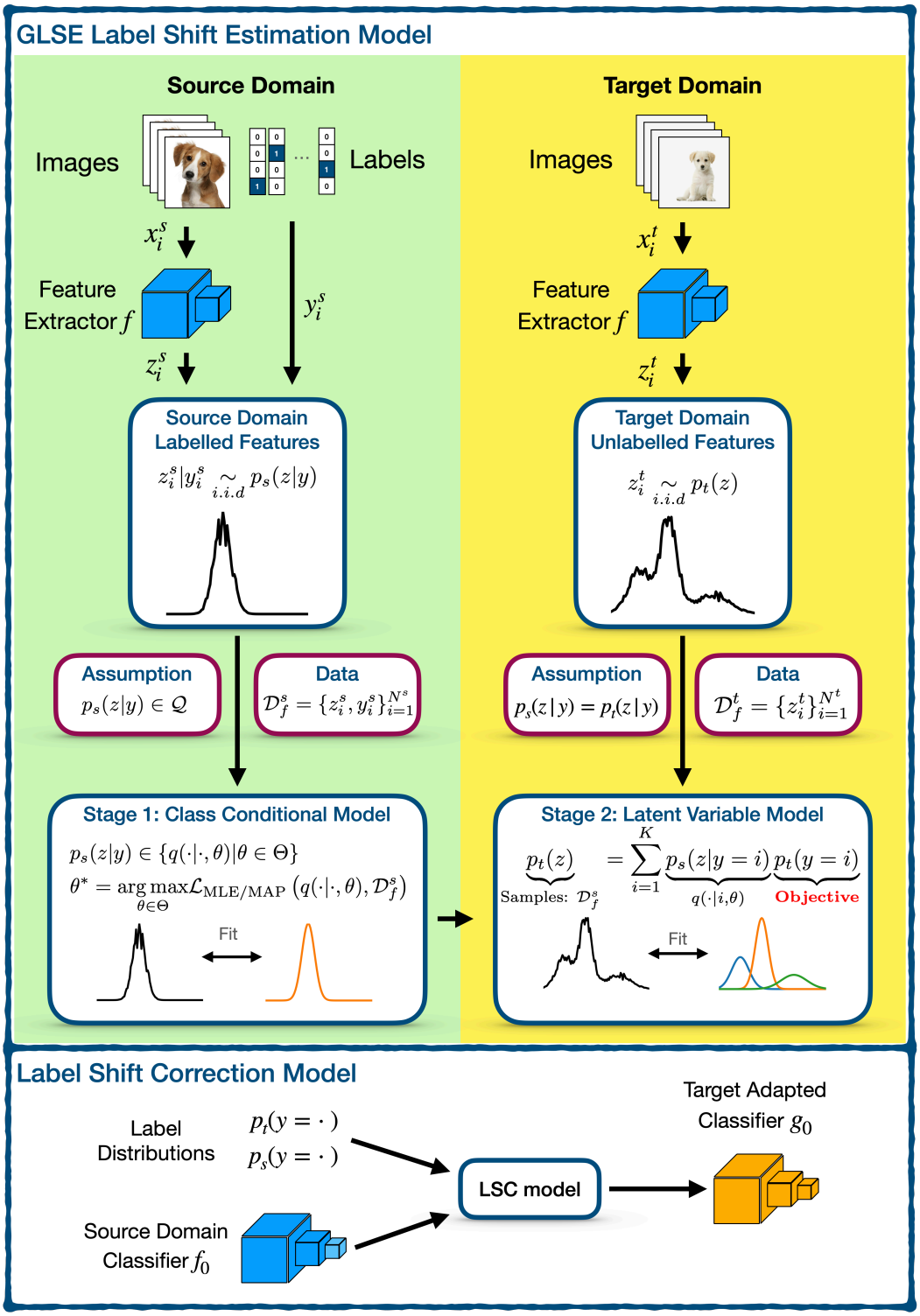

Figure 4.3: The structure of our GLSE label shift estimation model. In **Stage 1**, we assume $p_s(z|y) \in \mathcal{Q}$ and obtain MLE/MAP estimate θ for $p_s(z|y) = q(\cdot|\cdot,\theta)$ with the source domain dataset $\mathcal{D}_z^s$. In **Stage 2**, we estimate target label distribution $p_t(y=\cdot) = \boldsymbol{\pi}$ with a Latent Variable Model (LVM) given the target domain data $\mathcal{D}_z^t$. The estimation result can then be used for downstream tasks like label shift correction.

and Θ can then be defined as:

$$\begin{cases} q(z|j,\theta) = \dfrac{1}{B^j}\prod_{l=1}^{M} z_l^{\theta_l^j - 1} \\ \Theta = \left\{ \left[\theta^1, \theta^2, ...\theta^K\right] \middle| \theta^i \in \mathbb{R}^M_{>0} \right\}, \end{cases} \tag{4.5}$$

where B^j is the normalization constant of the Dirichlet distribution for class j. Θ is the space of parameters of K independent Dirichlet distributions. We use GLSE_{Dir} to denote the GLSE model with $\mathcal{Q}$ defined in Eq. (4.5).

SoftMax $\mathcal{Z}$ Space (Case 2): For $\mathcal{Z} = \Delta^{K-1}$, the SoftMax space has exactly the same dimension as the number of classes K in label space $\mathcal{Y}$. In this special case, we can model the distribution of $p_s(y|z)$ instead of $p_s(z|y)$:

$$p_s(y = j|z) = z_i. \tag{4.6}$$

$f : \mathcal{X} \to \Delta^{K-1}$ is said to be canonically calibrated on the source domain if the above equation is satisfied (Garg, Wu, Balakrishnan et al., 2020). MLLS (Saerens et al., 2002) adopts this model in their approach, as shown in Tabl. 4.1.

In this case, via Bayes rule, we can define the $q(\cdot|j,\theta)$ and parameter space Θ in $\mathcal{Q}$ as:

$$\begin{cases} q(z|j,\theta) = z/\theta_j \cdot p_s(z), \\ \Theta = \left\{ \theta \middle| \theta \in \Delta^{K-1} \right\}, \end{cases} \tag{4.7}$$

where $\theta := p_s(y) = \boldsymbol{c} \in \Delta^{K-1}$ is the parameter of the source label distribution $Y_s \sim \text{Cat}(K, \boldsymbol{c})$ and $p_s(z)$ is the marginal distribution of feature Z_s in the source domain.

The parameter $\boldsymbol{c}$ can be estimated given the source domain label in $\mathcal{D}_z^s$. The marginal distribution $p_s(z)$, however, can be intractable. Fortunately, in **Stage 2** of our model, we propose an algorithm that can estimate label shift without knowing $p_s(z)$. We use $\text{GLSE}_{\text{z/c}}$ to denote the GLSE model with $\mathcal{Q}$ defined in Eq. (4.5).

Neural Network Feature $\mathcal{Z}$ Space: A prototype NN classifier f usually uses the last convolutional layer output of a NN model to generate prototypes (Snell et al., 2017a). In this case, the features from each class can be modelled with a multivariate Gaussian distribution with a diagonal covariance matrix (Lee et al., 2018b; Morteza and Li, 2022). The $q(z|y,\theta)$ and Θ in $\mathcal{Q}$ can then be defined as:

$$\begin{cases} q(z|j,\theta) = \dfrac{1}{B^j}\exp\left(-\sum_{j=1}^{K}\dfrac{(z-\mu^j)^2}{2(\sigma^j)^2}\right) \\ \Theta = \left\{ (\mu^1,\sigma^1), ..., (\mu^K,\sigma^K) \middle| \mu^i \in \mathbb{R}^M, \sigma^i \in \mathbb{R}^M_{>0} \right\}, \end{cases} \tag{4.8}$$

where B^j is the normalization constant and μ_j, σ^j are mean and variances of the j^{th} Gaussian.

Here Θ is the parameter space for K Gaussian distributions where the j^{th} Gaussian has a mean μ^j and covariance matrix σ^j. $q(z|i,\theta)$ is the evaluation of probability density of the i^{th} Gaussian at z. We use $\text{GLSE}_{\mathcal{N}}$ to denote the GLSE model with $\mathcal{Q}$ defined in Eq. (4.8).

Parameter Estimate: With the distribution family $\mathcal{Q}$ specified, we can use the source domain labeled dataset $\mathcal{D}_z^s$ to obtain a point estimate of θ for $p_s(z|y) = q(\cdot|\cdot,\theta)$. A straightforward choice is MLE. If a prior distribution over θ is introduced, we can also obtain a MAP estimate of θ'. Whether to use the MLE or MAP estimate can be determined based on practical situations.

4.3.4 Stage 2: Latent Variable Model

In this section, given $\mathcal{Q}$ and a point estimate of θ' obtained from **stage 1**, we derive EM algorithms for the LVM in Eq. (4.3) to obtain MLE and MAP estimates of the target label distribution $p_t(y = \cdot) = \boldsymbol{\pi}$. We also adopt the Bayesian approach to obtain samples of posterior of $\boldsymbol{\pi}$ vian MCMC.

Maximum Likelihood Estimate If $p_s(z|y) = q(\cdot|\cdot,\theta)$, based on Eq. (4.3), we can write down the negative log likelihood of $\boldsymbol{\pi}$ given the target domain dataset $\mathcal{D}_z^t$ as:

$$-\log L(\boldsymbol{\pi}; \mathcal{D}_z^t) = -\log \prod_{i=1}^{N^t} \sum_{j=1}^{K} q(z_i^t|j,\theta)\pi_j. \tag{4.9}$$

Any MLE of $\boldsymbol{\pi}$ satisfies:

$$\boldsymbol{\pi}^{\text{MLE}} \in \underset{\boldsymbol{\pi}\in\Delta^{K-1}}{\arg\min} -\log L(\boldsymbol{\pi}; \mathcal{D}_z^t) \tag{4.10}$$

We prove that the MLE objective in Eq. (4.10) is convex and derive an EM algorithm to obtain $\boldsymbol{\pi}^{\text{MLE}}$ in the LVM (detailed proof available in Appendix B.1.1).

Proposition 3. ***(MLE)*** *Under Assumption 1, 4, if $p_s(z|y=i) = q(\cdot|i,\theta)$, then Eq. (4.9) is convex on $\boldsymbol{\pi}$ and EM algorithm 3 converges to a $\boldsymbol{\pi}^{MLE}$ defined in Eq. (4.10).*

The notation $\boldsymbol{\pi}^{(0)} \in \Delta_{>0}^{K-1}$ denotes $\boldsymbol{\pi}^{(0)} \in \Delta^{K-1}$ and $\pi_i > 0$ for all $i \in \{1, 2, ..., K\}$.

Maximum *a Posteriori* estimate By introducing a prior $\boldsymbol{\pi} \sim p(\boldsymbol{\pi}|\boldsymbol{\alpha})$ over parameter $\boldsymbol{\pi}$, similar to MLE, we can write the negative log posterior of $\boldsymbol{\pi}$ given the target domain feature $\mathcal{D}_z^t$ and prior parameter $\boldsymbol{\alpha}$ as:

$$-\log p(\boldsymbol{\pi}|\mathcal{D}_z^t, \boldsymbol{\alpha}) = -\log \frac{1}{C} p(\boldsymbol{\pi}|\boldsymbol{\alpha}) \prod_{i=1}^{N^t} \sum_{j=1}^{K} q(z_i^t|j,\theta)\pi_j, \tag{4.13}$$

where C is the normalization constant.

Algorithm 3 GLSE-MLE

Input: $\mathcal{D}_z^t = \{z_i^t\}_{i=1}^{N^t}, q(\cdot|\cdot,\theta) \in \mathcal{Q}$.
Initialize: $\boldsymbol{\pi}^{(0)} \in \Delta_{>0}^{K-1}$.
for $t = 0$ to M **do**
E-step: Evaluate

$$g_{ij}^{(m)} = \frac{\pi_j^{(m)} q(z_i^t|j,\theta)}{\sum_{l=1}^K \pi_l^{(m)} q(z_i^t|l,\theta)}. \tag{4.11}$$

M-step: Evaluate

$$\pi_j^{(m+1)} = \frac{1}{N^t} \sum_{i=1}^{N^t} g_{ij}^{(m)}. \tag{4.12}$$

end for
Output: $p_t(y = \cdot) = \boldsymbol{\pi}^{(m+1)}$.

Any MAP estimate of $\boldsymbol{\pi}$ satisfies:

$$\boldsymbol{\pi}^{\text{MAP}} \in \arg\min_{\boldsymbol{\pi} \in \Delta^{K-1}} -\log p(\boldsymbol{\pi}|\mathcal{D}_z^t, \boldsymbol{\alpha}) \tag{4.14}$$

In Proposition 3 we have shown that the negative log likelihood (Eq. (4.9)) is convex on $\boldsymbol{\pi}$. If the prior $p(\boldsymbol{\pi}|\boldsymbol{\alpha})$ is log strictly concave, the negative log posterior (Eq. (4.13)) will be strictly convex on $\boldsymbol{\pi}$ and $\boldsymbol{\pi}^{\text{MAP}}$ will be unique.

Lemma 4. *If the $p(\boldsymbol{\pi}|\boldsymbol{\alpha})$ in Eq. (4.13) is log strictly concave, then objective Eq. (4.13) is strictly convex on $\boldsymbol{\pi}$.*

Detailed proof is available in Appendix B.1.2.

Parameter $\boldsymbol{\pi} \in \Delta^{K-1}$ is a probability simplex, requiring the prior $p(\boldsymbol{\pi}|\boldsymbol{\alpha})$ to be supported on Δ^{K-1}. We propose to employ a Dirichlet prior over $\boldsymbol{\pi}$, as Dirichlet distributions over K dimensions satisfies this constraint and are log strictly concave (Joo et al., 2020; Tu, 2014). We derive an EM algorithm to obtain the unique MAP estimate $\boldsymbol{\pi}^{\text{MAP}}$ (detailed proof is available in Appendix B.1.3).

Proposition 5. ***(MAP estimate)*** *Under Assumption 1, 4 with $p_s(z|y=i) = q(\cdot|i,\theta)$. If parameter $\boldsymbol{\pi} \sim Dir(K, \boldsymbol{\alpha})$ with $\boldsymbol{\alpha} \in \mathbb{R}_{>1}^K$, then EM algorithm 4 converges to the $\boldsymbol{\pi}^{MAP}$ defined in Eq. (4.14).*

In the GLSE-MAP algorithm, the M-Step can be rewritten as the following form:

$$\pi_j^{(m+1)} = \lambda \underbrace{\frac{\sum_{i=1}^{N^t} g_{ij}^{(m)}}{N^t}}_{\text{Data contribution}} + (1-\lambda) \underbrace{\frac{\alpha_j - 1}{\sum_{l=1}^K (\alpha_l - 1)}}_{\text{Prior contribution}}, \tag{4.17}$$

where $\lambda = N^t / (N^t + \sum_{l=1}^K (\alpha_l - 1))$.

Algorithm 4 GLSE-MAP

Input: $\mathcal{D}_z^t = \{z_i^t\}_{i=1}^{N^t}$, $q(\cdot|\cdot,\theta) \in \mathcal{Q}$, $\boldsymbol{\alpha} \in \mathbb{R}_{>1}^K$.
Initialize: $\boldsymbol{\pi}^{(0)} \in \Delta_{>0}^{K-1}$.
for $t = 0$ to M **do**
E-step: Evaluate

$$g_{ij}^{(m)} = \frac{\pi_j^{(m)} q(z_i^t|j,\theta)}{\sum_{l=1}^K \pi_l^{(m)} q(z_i^t|l,\theta)}. \tag{4.15}$$

M-step: Evaluate

$$\pi_j^{(m+1)} = \frac{\sum_{i=1}^{N^t} g_{ij}^{(m)} + \alpha_j - 1}{N^t + \sum_{l=1}^K (\alpha_l - 1)}. \tag{4.16}$$

end for
Output: $p_t(y = \cdot) = \boldsymbol{\pi}^{(m+1)}$.

As $\lambda \to 1^-$, the algorithm degenerates to GLSE-MLE. As $\lambda \to 0^+$, the MAP estimate will converge to the Dirichlet prior $\mathrm{Dir}(K, \boldsymbol{\alpha})$. In this manner, λ can be seen as our confidence in our label distribution estimation.

Symmetric Dirichlet Prior: The hyperparameter $\boldsymbol{\alpha}$ in Dirichlet prior containing K elements to be determined in practice. To simplify the problem, we can let $\alpha_j = \alpha_0$ when no information about $\boldsymbol{\pi}$ is available a priori. This approach sacrifices the flexibility of the Dirichlet prior to reduce complexity of the model. In this case, the Dirichlet prior satisfies $\boldsymbol{\pi} \sim \mathrm{Dir}(K, \alpha_0 \mathbf{1})$.

Then the M-Step of the GLSE-MAP algorithm 4 can be rewritten as:

$$\pi_j^{(m+1)} = \lambda \frac{\sum_{i=1}^{N^t} g_{ij}^{(m)}}{N^t} + (1-\lambda)\frac{1}{K}, \tag{4.18}$$

where $\lambda = N^t / (N^t + K(\alpha_0 - 1))$.

The GLSE-MAP algorithm with $\lambda \to 0^+$ will converge to a uniform categorical distribution with $\boldsymbol{\pi} = \mathbf{1}/K$. Note that now $\boldsymbol{\alpha} = \alpha_0 \mathbf{1}$ is fully determined by λ, we can determine parameter α_0 in the prior by selecting a value for λ. In this case, $1 - \lambda$ represents the strength of regularization in the MAP estimation procedure.

Alternative Condition for LVM As shown in Eq. (4.7), we provide an example of $\mathcal{Q}$ that is constructed based on $p_s(y = i|z) = z_i$ instead of $p_s(z|y = i)$ directly. In this section, we prove that $\mathcal{Q}$ can be constructed as long as $p_s(y = i|z)$ is given.

We prove that if $p_s(y = i|z)$ and the source domain label distribution $p_s(y) = \boldsymbol{c}$ are given, our EM algorithms 3 and 4 are still able to converge to $\boldsymbol{\pi}^{\mathrm{MLE}}$ and $\boldsymbol{\pi}^{\mathrm{MAP}}$ respectively.

Proposition 6. *Under Assumption 1 and Assumption 4, let* $p_s(y=i|z) = h(i;\cdot)$ *and* $Y_s \sim Cat(K, \boldsymbol{c})$*, if we define:*

$$q(z_0|i,\theta) = h(i;z_0)/\theta_i \tag{4.19}$$

with $\theta := \boldsymbol{c} \in \Delta^{K-1}$*, then GLSE-MLE 3 converges to a* $\boldsymbol{\pi}^{MLE}$ *and GLSE-MAP 4 converges to the unique* $\boldsymbol{\pi}^{MAP}$*.*

Detailed proof can be found in Appendix B.1.4.

With Proposition 6, if we can obtain $p_s(y=i|z)$ in **Stage 1**, in stage 2 we are still able to obtain a point estimate of target label distribution $\boldsymbol{\pi}$ with EM algorithms. Similarly, we can still use MCMC to obtain samples from the posterior $P(\boldsymbol{\pi}|\mathcal{D}_z^t, \boldsymbol{\alpha})$ based on Eq. (4.19).

Proposition 6 also implies that the previous classifier based label shift estimation model MLLS (Saerens et al., 2002) and MAPLS proposed in Chapter 3 can both be seen as a special case of the GLSE model that obtains MLE and MAP estimate of target label distribution respectively with Eq. (4.19) satisfied.

Relation with Previous Models: As shown in Tab. 4.1, the proposed GLSE model in this chapter can be seen as a generalization of the MLLS model (Saerens et al., 2002) and the MAPLS model proposed in Chapter 3. Moreover, other previous model like BBSE and RLLS can be seen as having the same **Stage 1** estimation but different **Stage 2** estimation compare with our model. This similarity has been discussed from the perspective of BBSE by Garg, Wu, Balakrishnan et al. (2020).

Note that as a generalization of the MAPLS (Chapter 3) model, in GLSE model we can also use the previously proposed prior parameter learning (Adaptive Prior Learning) model that determines the parameter in the Dirichlet prior and the posterior sampling method that obtains i.i.d. samples from the posterior.

Models	Feature Extractor	Stage 1		Stage 2	
		Distribution Family $\mathcal{Q}$	θ Estimation	Algorithm	π Estimation
BBSE	$f: \mathcal{X} \to \{1,2...K\}$	$q(z\|j,\theta) = \theta_z^j$	MLE	solve linear system	
RLLS	$f: \mathcal{X} \to \{1,2...K\}$	$q(z\|j,\theta) = \theta_z^j$	MLE	solve constrained linear system	
MLLS	$f: \mathcal{X} \to \Delta^{K-1}$	$q(z\|j,\theta) = z/c_j$	MLE	EM	MLE
$\text{GLSE}_{\text{z/c}}$ **(ours)**	$f: \mathcal{X} \to \Delta^{K-1}$	$q(z\|j,\theta) = z/c_j$	MLE/MAP	EM	MLE/MAP/Posterior
GLSE_{Dir} **(ours)**	$f: \mathcal{X} \to \Delta^{M-1}$	$q(z\|j,\theta) = \text{Dir}(L,\theta^j)$	MLE/MAP	EM	MLE/MAP/Posterior
$\text{GLSE}_{\mathcal{N}}$ **(ours)**	$f: \mathcal{X} \to \mathbb{R}^L$	$q(z\|j,\theta) = \mathcal{N}(\mu^j,\sigma^j)$	MLE/MAP	EM	MLE/MAP/Posterior
GLSE_{Cat} **(ours)**	$f: \mathcal{X} \to \{1,2...L\}$	$q(z\|j,\theta) = \theta_z^j$	MLE/MAP	EM	MLE/MAP/Posterior

Table 4.1: Structure comparison of different label shift estimation models. With different choices of feature extractor f, distribution family $p_s(z|y) \in \mathcal{Q}$ in **Stage 1** and different estimation methods in **Stage 2**, a variety of label shift estimation models can be create under our GLSE framework. MLLS can be seen as a special case of our $\text{GLSE}_{\text{z/c}}$ model.

4.3.5 Practical Example of GLSE model

We provide a simple example of a practical label shift estimation model GSLE_{Cat}-MAP that is created under our GLSE framework. The model adopts the following

settings:

- Feature extractor $f : \mathcal{X} \to \{1, 2, ..., M\}$.
- **Stage 1:** Define $\mathcal{Q}$ with Eq. (4.4), obtain MLE of θ in $p_s(z|y = i) = q(\cdot|i, \theta)$.
- **Stage 2:** Use algorithm GLSE-MAP 4 to obtain $\boldsymbol{\pi}^{\text{MAP}}$.

In this case, our two-stage GLSE framework is written as:

Example 7. *For $f : \mathcal{X} \to \{1, 2, ..., L\}$, under Assumption 1, Assumption 4 with $\mathcal{Q}$ defined in Eq. (4.4), Alg. 5 converges to the $\boldsymbol{\pi}^{MAP}$.*

Algorithm 5 GLSE_{Cat}-MAP

Stage 1:

Input: $\mathcal{D}_z^s = \{z_i^s, y_i^s\}_{i=1}^{N^s}$.
Define $\mathcal{Q}$ with Eq. (4.4).
Obtain MLE of θ for $p_s(z = i|y = j) = \theta_i^j$:

$$\left(\theta^{\text{MLE}}\right)_i^j = \frac{1}{N^s} \sum_{l=1}^{N^s} \mathbb{I}_i(z_l^s)\mathbb{I}_j(y_l^s),$$

Stage 2:

Input: $\mathcal{D}_z^t = \{z_i^t\}_{i=1}^{N^t}, \theta^{\text{MLE}}$ and $\boldsymbol{\alpha} \in \mathbb{R}_{>1}^K$.
Initialize: $\boldsymbol{\pi}^{(0)} \in \Delta_{>0}^{K-1}$.
for $t = 0$ to M **do**

$$\pi_j^{(m+1)} = \lambda \sum_{i=1}^{L} \frac{b_i \left(\theta^{\text{MLE}}\right)_i^j \pi_j^{(m)}}{\sum_{l=1}^K \left(\theta^{\text{MLE}}\right)_i^l \pi_l^{(m)}} + (1 - \lambda)\frac{\alpha_j - 1}{\sum_{l=1}^K (\alpha_l - 1)},$$

where $b_i = \frac{1}{N^t} \sum_{l=1}^{N^t} \mathbb{I}_i(z_l^t)$ and $\lambda = N/(N^t + \sum_{l=1}^K (\alpha_l - 1))$.
end for
Output: $\boldsymbol{\pi} = \boldsymbol{\pi}^{(m+1)}$.

$\mathbb{I}(\cdot)$ is the indicator function and detailed derivation is available in Appendix B.1.5.

4.3.6 Overall Framework

In the GLSE framework, we first obtain the estimated $p_s(z|y = i)$ with the source domain dataset in **Stage 1**. Then the target label distribution $p_t(y = \cdot) = \boldsymbol{\pi}$ is estimated with the target domain dataset in **Stage 2**. Finally, we correct label shift of a source domain classifier f_0 with Eq. (2.34). The whole model is summarized in Algorithm 6.

Remark about the source label distribution $p_s(y = \cdot)$: MLE is the standard method to estimate the source domain label distribution $p_s(y = \cdot) = \boldsymbol{c}$ with ground truth labels $y_i^s \in \mathcal{D}_z^s$. When classifier f is canonically calibrated on the source domain, Alexandari et al. (2020) shows that it is also valid to estimate $\boldsymbol{c}$ with $c_j =$

Algorithm 6 GLSE Framework

Input: Target domain $\mathcal{D}^t$, Source domain $\mathcal{D}^s$ and feature extractor f, classifier f_0 based on f.
Assumptions: Assumption 1 and Assumption 4
Stage 1: On source domain
Input: $\mathcal{D}_z^s = \{(z_i^s, y_i^s) | z_i^s = f(x_i^s), (x_i^s, y_i^s) \in \mathcal{D}^s\}$,
1. Define $q(z|y,\theta)$ and Θ in $\mathcal{Q}$ based on f,
2. Obtain θ^{MLE} or θ^{MAP} for $p_s(z|y) \in \mathcal{Q}$.

Stage 2: On target domain
Input: $\mathcal{D}_z^t = \{z_i^t | z_i^t = f(x_i^t), x_i^t \in \mathcal{D}^t\}$
1. Construct LVM in Eq. (4.3).
2. Learn $\boldsymbol{\alpha}$ with APL model in Chapter 3.3.4 if $\boldsymbol{\pi} \sim \text{Dir}(K, \boldsymbol{\alpha})$.
3. Obtain $\boldsymbol{\pi}^* = \boldsymbol{\pi}^{\text{MLE}}$ or $\boldsymbol{\pi}^{\text{MAP}}$ with Alg. 3, 4 or samples drawn from $p(\boldsymbol{\pi}|\mathcal{D}^t, \boldsymbol{\alpha})$ with HMC.

Label Shift Correction: Correct classifier f_0 for label shift with $\boldsymbol{\pi}^*$ and $p_s(y) = \boldsymbol{c}$ based on Eq. (2.34).
Output: Label Shift Corrected Classifier $g_0(x)$

$\frac{1}{N^s}\sum_{i=1}^{N^s} f(x_i^s)_j$. We use "-s" to denote the model that uses the averaging approach and "-h" to denote the method that uses the MLE approach to estimate $\boldsymbol{c}$.

4.4 Experiments

4.4.1 Experimental Setup

Datasets: We evaluate our model on the CIFAR100 (Krizhevsky, Hinton et al., 2009), ImageNet 2012 (Russakovsky et al., 2015b) and Places2 (Zhou, Lapedriza et al., 2017) datasets. Following common use in Long-Tailed research (Cao et al., 2019; Wang, Lian et al., 2020; Zhong et al., 2021), we also use Long-Tailed versions of each dataset. We test the models on test sets with Dirichlet shift proposed by previous label shift estimation models (Alexandari et al., 2020; Lipton et al., 2018). Dirichlet Shift generates a random test set label distribution from a K dimensional Dirichlet distribution. We also adopt the ordered Long-Tailed shifted test set used in LADE (Hong, Han et al., 2021), which has the same or inverse order of the Long-Tailed distributed train set. We further extend this setting to a shuffled Long-Tailed test set, where the test set still has a Long-Tailed label distribution but with random order compared with the train set.

Remark: The label shift problem setup (Definition 1) requires the source and target data samples to be drawn i.i.d. from the data distributions of the corresponding domains. Therefore the rotation/translation invariance is naturally expected when these datasets contain real world data samples. As a result, common data augmentation techniques like random translation, flipping or color jittering can be used to preprocess the image samples before applying label shift models. No image feature selection technique is needed.

Model Setup: For all GLSE models, we initialized the EM algorithms with $\boldsymbol{\pi}^{(0)} = \boldsymbol{c}$ and ran for 100 epochs to ensure convergence. We follow the APL model setup in

Chapter 3 to determine the prior parameter. For our posterior sampling model, use a HMC sampler called No-U-Turn Sampler (Hoffman, Gelman et al., 2014) provided by Pyro (Bingham et al., 2018) to collect 5000 samples from the posterior. We implement the Neural Network classifiers using PyTorch (Paszke et al., 2017). We use the ResNet32 (Idelbayev, n.d.) classifier for the CIFAR100 and every CIFAR100-LT dataset. We use pre-trained ResNet50 (He, Zhang et al., 2016) and pre-trained Resnet152 for the ImageNet and Places datasets respectively. We train a ResNet50 and ResNet152 for the ImageNet-LT and Places-LT datasets respectively. Detailed implementations are available in Appendix B.2.1.

	Dataset	**Setup**
Train Set	CIFAR100 (Krizhevsky, Hinton et al., 2009)	Original, Long-Tailed with $R = \{10, 100\}$
	ImageNet (Russakovsky et al., 2015b)	Original, Long-Tailed
	Places (Zhou, Lapedriza et al., 2017)	Original, Long-Tailed
Test Set	**Test Shift Type**	**Params**
	Original	None
	Dirichlet (Lipton et al., 2018)	$\alpha = 1.0, 10$
	Ordered Long-Tailed(Hong, Han et al., 2021)	$R = \{2, 5, 10, 25, 50\}$ Order = "forward", "backward"
	Shuffled Long-Tailed	$R = \{2, 5, 10, 25, 50\}$

Table 4.2: Closed SEt Label Shift experiment setups. R is referred to as the imbalance ratio — the ratio of maximum and minimum sample number per class respectively in the test set. α is the parameter of the Dirichlet distribution.

Evaluation Metrics: We follow previous methods (Alexandari et al., 2020; Lipton et al., 2018) to evaluate the performance of label shift estimation models with $(\boldsymbol{w} - \hat{\boldsymbol{w}})^2/K$, where $w_i = \pi_i/c_i$ is the target over the source label distribution ratio, with $\boldsymbol{w}$ as the ground truth and $\hat{\boldsymbol{w}}$ as the prediction. We also provide Top1 accuracy of different label shift estimation models with offline LSC on all datasets.

Remark: In the Figs in this section, the "**Amount of label shift**" is measured by the KL divergence between the ground truth target and source label distribution $D_{\mathrm{KL}}(\boldsymbol{\pi}\|\boldsymbol{c})$, which is normalized in range $[0, 1]$ by the maximal possible value of the label shift settings under consideration in Tab. 4.2.

4.4.2 State-of-the-art Comparison

We compare the performance of our method with several state-of-the-art (SOTA) label shift estimation methods, including MLLS (Alexandari et al., 2020; Saerens et al., 2002), BBSE (Lipton et al., 2018) and RLLS (Azizzadenesheli et al., 2018). BBSE and RLLS also have "soft" and "hard" versions of each model. We evaluate the performance of these models with previously available implementations; details are given in Appendix B.2.2. In large-scale datasets, methods that require retraining the classifier on the source domain will suffer from high computational costs. Therefore, we have not reproduced and reported Tachet des Combes et al. (2020) in our results.

Train Set \ Test Set	Ordered LT	Shuffled LT	Dirichlet
CIFAR100/CIFAR100-LT	52%	43%	60%
Places/Places-LT	68%	80%	75%
ImageNet/ImageNet-LT	77%	90%	75%

Table 4.3: SOTA comparison summary of estimation error. For all the label shift settings in Tab. 4.2, the percentage of settings that our $\text{GLSE}_{\text{z/c}}$/GLSE_{Cat} model outperforms SOTA models in terms of $(w - \hat{w})^2/K$.

Train Set \ Test Set	Ordered LT	Shuffled LT	Dirichlet
CIFAR100/CIFAR100-LT	66%	83%	63%
Places/Places-LT	73%	90%	92%
ImageNet/ImageNet-LT	82%	100%	75%

Table 4.4: SOTA comparison summary of Top1 Accuracy. For all the label shift settings in Tab. 4.2, the percentage of settings that our $\text{GLSE}_{\text{z/c}}$/GLSE_{Cat} model outperforms SOTA models in terms of Top 1 accuracy.

We compare the performance of our $\text{GLSE}_{\text{z/c}}$ and GLSE_{Cat} model with SOTA models, as $\text{GLSE}_{\text{z/c}}$ and MLLS have almost identical $\mathcal{Q}$ setup and GLSE_{Cat} and BBSE/RLLS have almost identical $\mathcal{Q}$ setup. The results summary is provided in Tab. 4.3 and Tab. 4.4. We also provide some detailed results in Tab. 4.5, Tab. 4.6 and Fig. 4.4. In Fig. 4.4, we report the label shift estimation error $(w - \hat{w})^2/K$ for all datasets under Ordered-LT setting. The horizontal axis "**Amount of label shift**" is measured by the KL divergence between the ground truth target and source label distribution $D_{\text{KL}}(\pi \| c)$, which is normalized in range $[0, 1]$ by the maximal possible value in the Ordered-LT test set settings given in Tab. 4.2.

As shown in Fig. 4.4, all the tested models exhibit higher estimation errors under larger label shift. Compared with other models, the performance of our GLSE models with MAP estimate (*e.g.*"GLSE_{Cat}-MAP-APL") exhibit a "U" curve. This is because these models introduce a Dirichlet prior that helps regularize the estimation, which can improve performance under large label shift settings at the cost of performance loss in the small label shift settings.

As seen from Tab. 4.5,4.6, 4.7, 4.8 and Fig. 4.4, one obvious advantage of our model is its robustness to the highly imbalanced label distribution. Our model outperforms previous models in most settings when the source domains have a highly imbalanced label distributions (*e.g.* CIFAR100-100-LT, ImageNet-LT, Places-LT) or the amount of label shift between source and target domain is larges.

4.4.3 GLSE model selection

We proposed four practical GLSE models for different feature extractors f, and each model admits both MLE and MAP estimates of target label distribution (Tab. 4.1). This section provides experiments to compare the performance of the different GLSE models on the CIFAR100, CIFAR100-10-LT and CIFAR100-100-LT datasets.

Dataset	ImageNet								
Order	Forward				Uniform	Backward			
Imbalance Ratio	25	10	5	2	1	2	5	10	25
Baseline	78.05	77.73	77.35	**76.73**	76.13	75.48	74.60	73.89	73.05
MLLS-h	77.95	77.09	76.38	75.54	75.03	74.40	73.97	73.83	73.79
MLLS-s	78.11	77.31	76.59	75.79	75.27	74.67	74.34	74.20	74.19
BBSE-h	78.01	77.29	76.59	75.86	75.33	74.70	74.26	74.09	73.98
BBSE-s	77.86	77.16	76.52	75.74	75.21	74.60	74.22	74.03	73.89
RLLS-h	78.05	77.73	77.35	76.73	76.13	75.48	74.60	73.89	73.05
RLLS-s	78.05	77.73	77.35	76.73	76.13	75.48	74.60	73.89	73.05
$\text{GLSE}_{z/c}$-MAP-s (**ours**)	78.36	77.59	76.87	76.06	75.50	74.93	74.56	74.42	74.37
$\text{GLSE}_{z/c}$-MAP-APL-s (**ours**)	78.36	77.59	76.88	76.10	75.57	74.97	74.57	74.42	74.37
$\text{GLSE}_{z/c}$-PS-APL-s	78.08	77.31	76.68	76.09	75.52	74.89	74.62	74.17	74.25
GLSE_{Cat}-MLE-s (**ours**)	77.47	76.86	76.35	75.64	75.14	74.50	74.02	73.73	73.31
GLSE_{Cat}-MAP-s (**ours**)	**78.68**	77.90	77.23	76.35	75.76	75.17	74.74	74.53	**74.52**
GLSE_{Cat}-MAP-APL-s (**ours**)	78.67	**77.97**	**77.40**	76.71	**76.15**	**75.51**	**74.89**	**74.59**	74.51

Table 4.5: Top1 Acc on ImageNet dataset (20 run average), with Ordered Long-Tailed test set that have imbalance ratio $R = \{25, 10, 5, 2\}$. Our GLSE_{Cat} model outperforms other models on the ImageNet dataset.

Which GLSE model As shown in Fig. 4.5, our $\text{GLSE}_{z/c}$ model performs better than other GLSE models in most of the tested settings. Therefore, we recommend that the user to choose the $\text{GLSE}_{z/c}$ model if applicable.

MLE vs MAP vs MAP-APL As shown in Fig. 4.5, MAP-APL models perform better under large label shift settings, while MLE models perform better under small label shift settings. Generally, when the GLSE model is fixed, MAP-APL outperforms MLE/MAP in most settings.

"hard" vs "soft" As shown in Fig. 4.6, the soft version of GLSE models tend to perform better under large label shift settings, while hard versions of GLSE models tend to perform better under small label shift settings.

Based on the experimental results, we recommend the user to select GLSE models with the following rules: (1) Use $\text{GLSE}_{z/c}$ model if applicable. (2) Use GLSE-MAP-APL-s model in the large label shift settings and GLSE-MLE-h in the small label shift settings. (3) Use the GLSE-PS model when the uncertainty of the estimation is essential.

4.4.4 Model Calibration Performance

A calibrated classifier can provide accurate prediction and the confidence of the prediction (Guo, Pleiss et al., 2017). In the label shift problem, we also hope the label shift corrected classifier created based on the label shift estimation models are well calibrated. This chapter analyzes the calibration performance of the classifiers based on our model and previous models with Expected Calibration Error (ECE) (Podkopaev and Ramdas, 2021). As shown in Fig. 4.7, our $\text{GLSE}_{z/c}$-MAP-APL model outperform previous models and other GLSE models in most of the settings on Places/Places-LT datasets. Our GLSE models generally perform better on the Places-LT dataset.

Dataset	ImageNet-LT								
Order	Forward				Uniform	Backward			
Imbalance Ratio	25	10	5	2	1	2	5	10	25
Baseline	**62.52**	**58.50**	54.96	49.65	45.31	41.02	35.23	31.28	26.70
MLLS-h	59.22	55.53	52.67	48.98	46.47	44.10	41.28	39.70	38.38
MLLS-s	58.42	54.81	52.18	48.52	46.14	44.24	41.58	40.34	39.46
BBSE-h	26.62	32.17	29.18	27.73	15.11	26.36	11.65	27.01	21.85
BBSE-s	61.11	57.52	54.82	51.11	48.29	45.63	42.35	40.21	37.62
RLLS-h	62.52	58.50	54.96	49.65	45.31	41.02	35.23	31.28	26.70
RLLS-s	62.52	58.50	54.96	49.65	45.31	41.02	35.23	31.28	26.70
$\text{GLSE}_{z/c}$-MAP-s (**ours**)	60.11	56.90	54.39	50.99	48.56	46.65	44.24	43.02	41.78
$\text{GLSE}_{z/c}$-MAP-APL-s (**ours**)	60.42	57.63	55.44	52.55	50.35	**48.26**	**45.63**	**44.05**	**42.51**
$\text{GLSE}_{z/c}$-PS-APL-s	60.61	57.96	55.45	**52.77**	**50.45**	48.00	45.46	43.92	42.21
GLSE_{Cat}-MLE-s (**ours**)	60.17	56.36	53.13	48.70	45.30	41.88	37.17	34.24	30.45
GLSE_{Cat}-MAP-s (**ours**)	61.60	58.25	55.53	51.81	48.91	45.92	41.93	39.28	35.98
GLSE_{Cat}-MAP-APL-s (**ours**)	61.56	58.38	**55.90**	52.49	49.88	47.23	43.43	40.80	37.47

Table 4.6: Top1 Acc on ImageNet-LT dataset (20 run average), with Ordered Long-Tailed test set that have imbalance ratio $R = \{25, 10, 5, 2\}$. Our $\text{GLSE}_{z/c}$ model outperform other models in the "backward" settings on the ImageNet-LT dataset.

Order	Forward					Uniform	Backward				
Imbalance Ratio	50	25	10	5	2	1	2	5	10	25	50
Baseline	55.37	55.66	56.08	56.21	56.59	**56.77**	56.91	56.88	56.96	56.97	56.85
MLLS-h	**60.67**	**59.16**	57.49	56.31	55.52	55.53	55.94	57.05	58.41	60.51	61.88
MLLS-s	60.65	59.16	57.65	56.48	55.69	55.59	55.98	57.07	58.45	**60.59**	**62.05**
BBSE-h	55.54	54.34	52.80	51.55	50.78	50.77	51.40	52.59	54.21	56.16	57.79
BBSE-s	60.05	58.62	57.12	55.93	55.19	55.06	55.33	56.37	57.87	60.07	61.55
RLLS-h	55.37	55.66	56.08	56.21	56.59	56.77	56.91	56.88	56.96	56.97	56.85
RLLS-s	55.37	55.66	56.08	56.21	56.59	56.77	56.91	56.88	56.96	56.97	56.85
$\text{GLSE}_{z/c}$-MAP (**ours**)	60.29	59.05	**57.83**	56.82	56.17	56.18	56.57	57.46	58.74	60.57	61.83
$\text{GLSE}_{z/c}$-MAP-APL (**ours**)	60.29	59.05	57.83	56.81	56.20	56.23	56.61	57.48	58.74	60.55	61.82
GLSE_{Cat}-MLE (**ours**)	58.48	57.19	55.90	54.99	54.48	54.56	54.63	55.42	56.61	58.37	59.68
GLSE_{Cat}-MAP (**ours**)	59.50	58.60	57.66	**56.83**	56.43	56.38	56.83	**57.66**	**58.78**	60.23	61.17
GLSE_{Cat}-MAP-APL (**ours**)	59.16	58.26	57.44	56.79	**56.66**	56.73	**57.01**	57.63	58.55	60.00	60.87

Table 4.7: Top1 Accuracy on the Places dataset (20 run average), with Ordered Long-Tailed test sets that have imbalance ratio $R = \{50, 10, 5, 2\}$. Each reported value averages over 20 different shuffled and random sampled test sets.

4.4.5 Empirical Analysis on Computational Complexity

We analyze the computational complexity of our model through monitoring the training time of our model on dataset with different number of classes. The average training time of each label shift estimation models are provided in Tab. 4.9.

As shown in Tab. 4.9, the training time of our algorithm and other label shift estimation models is within the scale of seconds and increase linearly w.r.t. the number of classes in the dataset. Since the training time of our $\text{GLSE}_{z/c}$/GLSE_{Cat} model on the ImageNet dataset with 1000 classes is less than 1 minute, and the average training time of a ResNet50 NN classifier on the entire ImageNet dataset with the same hardware setup can last for more than 24 hours, we believe such model can be applied to large scale dataset with negligible computational overhead.

Order	Forward					Uniform	Backward				
Imbalance Ratio	50	25	10	5	2	1	2	5	10	25	50
Baseline	**43.47**	41.34	37.84	35.09	31.04	27.92	24.85	20.88	18.28	15.33	13.47
MLLS-h	42.33	40.45	37.63	35.61	33.00	31.08	29.29	26.79	25.29	23.62	22.59
MLLS-s	41.84	39.98	37.12	35.04	32.41	30.65	28.90	26.60	25.16	23.66	22.70
BBSE-h	31.41	28.92	28.79	24.09	19.68	24.31	17.94	22.84	22.48	18.72	20.13
BBSE-s	42.93	41.16	38.21	36.10	33.09	30.90	28.92	25.95	24.09	21.93	20.66
RLLS-h	43.47	41.34	37.84	35.09	31.04	27.92	24.86	20.88	18.28	15.33	13.47
RLLS-s	43.47	41.34	37.84	35.09	31.04	27.92	24.85	20.88	18.28	15.33	13.47
$\text{GLSE}_{z/c}$-MAP (**ours**)	42.42	40.74	38.27	36.47	34.07	32.36	30.76	28.48	27.00	25.34	24.32
$\text{GLSE}_{z/c}$-MAP-APL (**ours**)	42.52	41.16	39.13	**37.82**	**36.00**	**34.56**	**33.14**	**30.81**	**29.16**	**27.10**	**25.79**
GLSE_{Cat}-MLE (**ours**)	42.57	40.28	37.00	34.63	31.19	28.57	26.17	22.84	20.67	18.30	16.55
GLSE_{Cat}-MAP (**ours**)	43.39	**41.66**	39.01	37.18	34.32	32.27	30.28	27.42	25.40	22.94	21.48
GLSE_{Cat}-MAP-APL (**ours**)	43.26	41.63	**39.21**	37.55	35.00	33.07	31.26	28.54	26.50	23.91	22.46

Table 4.8: Top1 Accuracy on the Places-LT dataset (20 run average), with Ordered Long-Tailed test sets that have imbalance ratio $R = \{50, 10, 5, 2\}$. Each reported value averages over 20 different shuffled and random sampled test sets.

Model	CIFAR100	ImageNet	Places
MLLS	< 1	~ 20	~ 10
BBSE	< 1	~ 10	~ 2
RLLS	< 1	~ 200	~ 20
$\text{GLSE}_{z/c}$ (**ours**)	< 1	~ 50	~ 15
GLSE_{Cat} (**ours**)	< 1	~ 9	~ 7
GLSE_{Dir} (**ours**)	~ 40	-	-
$\text{GLSE}_{\mathcal{N}}$ (**ours**)	< 1	-	-

Table 4.9: Average training time comparison (seconds). Training time is compared on a NVIDIA RTX 2080Ti GPU. Our $\text{GLSE}_{z/c}$/GLSE_{Cat} model have comparable training time with SOTA models

4.5 Conclusion

In this chapter, 1) we propose a novel two-stage GLSE label shift estimation model for large-scale datasets and highly imbalanced source label distributions; 2) we construct four practical label shift estimation algorithms $\text{GLSE}_{z/c}$, GLSE_{Cat}, GLSE_{Dir}, $\text{GLSE}_{\mathcal{N}}$ under GLSE framework (Tab. 4.1); 3) we derive two EM algorithms to obtain a MLE/MAP estimate of the target label distribution (Alg. 3, 4) and discuss the convexity of MLE/MAP objectives; 4) we show that MLLS (Saerens et al., 2002) can be seen as a special case of our GLSE model (Tab. 4.1).

Experiments on CIFAR100, ImageNet, Places and Long-Tailed versions of each dataset show that our model consistently outperforms existing label shift estimation models, especially when the source or target domain have highly imbalanced label distributions. We analyze different GLSE models and provide model selection recommendations based on the experiments. Our model is robust to highly imbalanced source label distributions and has the potential to be applied in real world label shift problems.

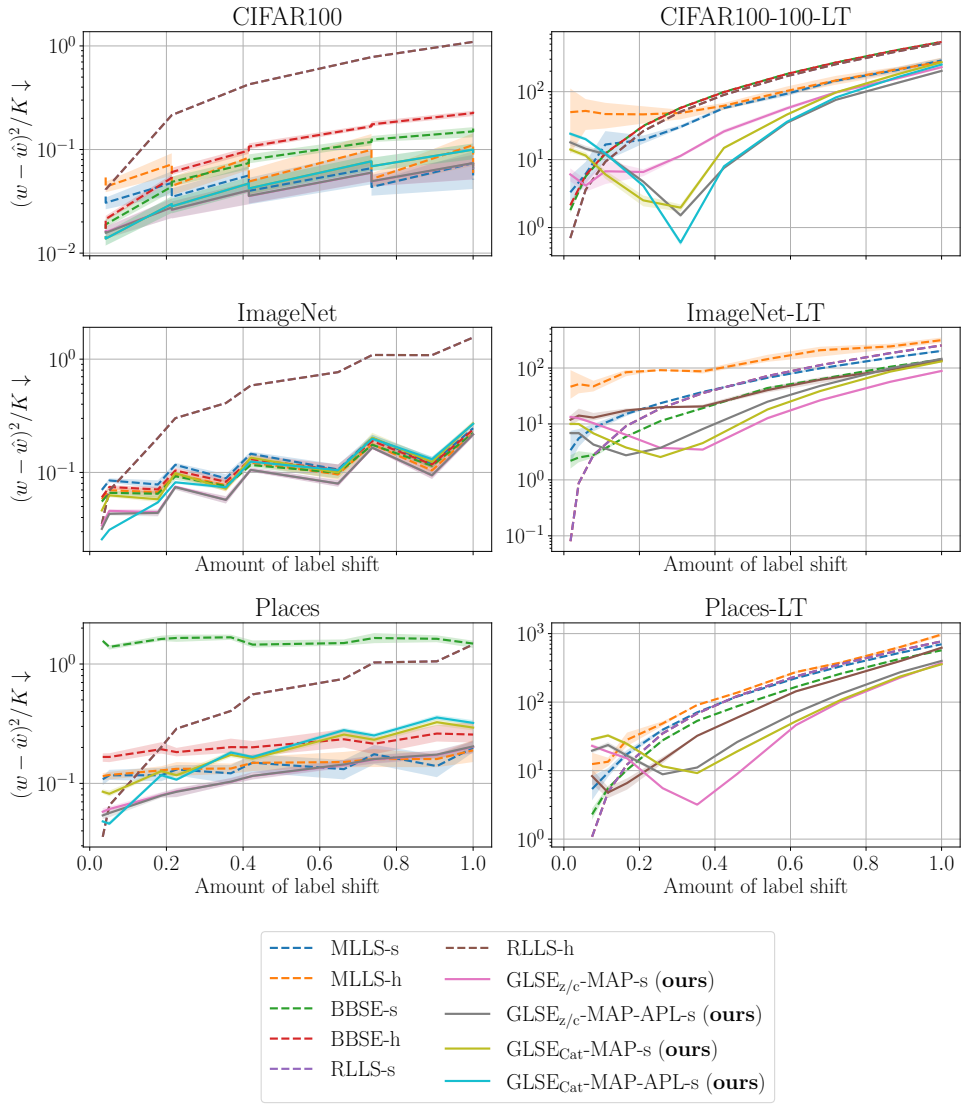

Figure 4.4: Estimation error of our GLSE models (solid lines) and previous models (dashed lines). Our models are robust under (a) large label shift and (b) highly class-imbalanced source domain datasets. The horizontal axis "Amount of label shift" is measured by the KL divergence between the ground truth target over source label distribution, *i.e.* $D_{\mathrm{KL}}(p_t(y = \cdot)\|p_s(y = \cdot))$, normalized to $[0,1]$ by the maximal possible value in the Ordered-LT test set settings given in Tab. 4.2. The label shift estimation error $(w - \hat{w})^2/K$ of existing models tends to increase as the amount of label shift increases or the source domain train set becomes highly class-imbalanced. Our GLSE$_{z/c}$ and GLSE$_{Cat}$ model outperform existing models in most of the reported Long-Tailed settings. The solid/dashed lines represent average performance, and shaded regions cover the minimal to maximal performance in 20 independent runs.

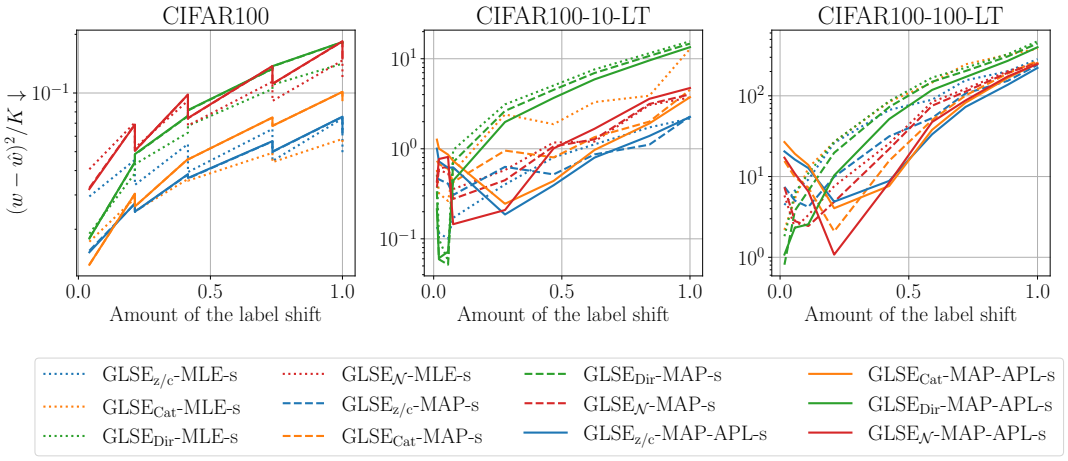

Figure 4.5: Estimation error of GLSE models with MLE (dotted lines), MAP with fixed $\lambda = 0.9$ (dashed lines) and MAP-APL model (solid lines). In general, our GLSE-MAP-APL models perform better than GLSE-MAP (fixed λ) models or GLSE-MLE models under large label shift. Our GLSE$_{z/c}$ model (blue) performs better than GLSE$_{Cat}$/GLSE$_{Dir}$/GLSE$_{\mathcal{N}}$ models.

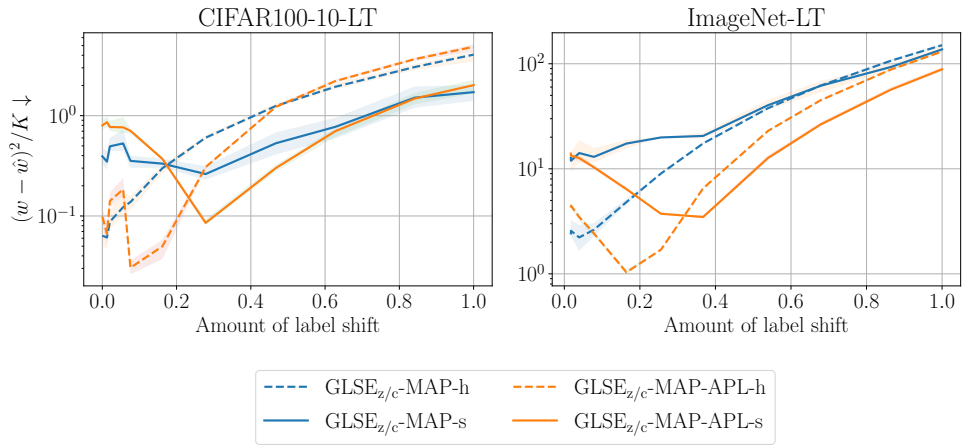

Figure 4.6: Estimation error of GLSE model with hard ("-h") and soft ("-s") versions. The soft version of the GLSE model performs better under large label shift settings, while the hard version performs better under small label shift settings.

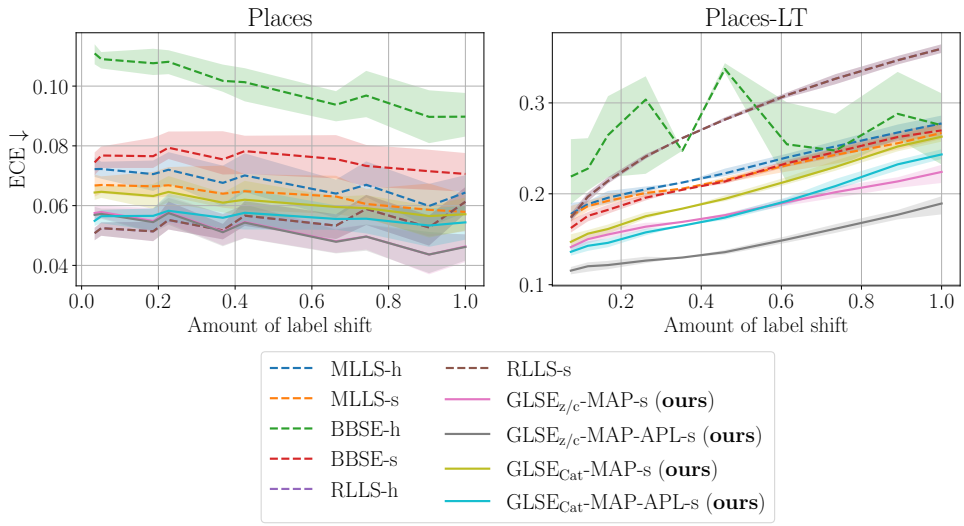

Figure 4.7: Calibration performance (ECE) comparison. Our GLSE models (solid lines) outperform previous models (dashed lines) on Place-LT datasets. Our GLSE$_{z/c}$-MAP-APL model outperforms other models in most settings.

Chapter 5

Classifier Based Open Set Label Shift

5.1 Introduction

Moving further from the Closed Set Label Shift problem discussed in Chapter 3 and Chapter 4, this chapter investigates the more challenging Open Set Label Shift (OSLS) problem. The OSLS problem extends the Closed Set Label Shift (CSLS) problem by introducing an extra out-of-distribution (OOD) class in the target domain. Since the source domain only has available samples from the in-distribution (ID) classes, the OSLS models must tackle the OOD samples without labeled OOD data in both the source and target domain. This chapter focuses mainly on the OSLS *estimation* problem.

Refresher on the OSLS problem In the *Open Set Label Shift* (OSLS) problem, we are given a source domain labeled dataset, a target domain unlabeled dataset, a source domain ID classifier and a source domain ID/OOD binary classifier. The objectives include 1) *detection*: verify the label distributions among the ID classes are identical between the source and target domain, 2) *estimation*: estimate the target label distribution for both ID classes and the extra OOD class and 3) *correction*: construct an appropriate target domain classifier. The formal definition of the OSLS problem is restated in the next section (Section 5.2.1), which is originally given in Section 2.1.

Motivations Our research on the OSLS problem is motivated by the following observations: 1) the OSLS problem is a common problem in the real world classification tasks, yet few models have been proposed and 2) the OSLS problem extends the Closed Set Label Shift (CSLS) problem to the open set setting. Therefore we can leverage the advances in the CSLS literature and the open set classification/out-of-distribution detection literature to tackle the problem.

Contributions This Chapter proposes a novel method for the OSLS estimation and correction problem. It uses an ID classifier and an ID/OOD classifier without retraining or fine-tuning. The ID/OOD classifier can be imported from the vast suite of methods available in the OOD detection/Open-Set Recognition literature. We derive EM algorithms for the Maximum Likelihood Estimate (MLE) or Maximum *a Posteriori* (MAP) estimate of the target label distribution and target ID data ratio with the OOD reference dataset. We also propose models to estimate the source ID data

ratio and target ID data ratio. We test the model on several datasets and show superior performance over state-of-the-art (SOTA) models.

Our main contributions are as follows:

- Based on a test time OOD dataset as a reference, we propose a novel OSLS model that estimates and corrects Open Set Label Shift without retraining the ID classifier and ID/OOD classifier. Our method can utilize existing OOD detection works without re-training or fine-tuning.
- We derive an EM algorithm to obtain the MLE/MAP estimate of the target ID label distribution and target ID data ratio (Theorem 10).
- We propose estimators for the source ID data ratio (which our EM algorithm requires) and the target ID data ratio for an imperfect OOD classifier. Upper bounds of the sampling error for the two estimators are also provided (Theorem 8 and 11).
- Experimental results demonstrate the superior performance of our method on both label shift estimation error in ID classes and label shift correction accuracy over baselines on CIFAR10/100 and ImageNet-200 datasets with various OOD datasets (Section 5.4).

Background and Related Works The background information about the EM algorithms can be found in Chapter 2, Section 2.2.5. The related works on the OSLS problem and OOD detection are given in Chapter 2, Section 2.3.1 and Section 2.3.4.

5.2 Problem Setup and Analysis

This section introduces the notations used in this Chapter, provides a refresher on the Open Set Label Shift problem we consider, and introduces the assumptions used to construct our model.

Let $\mathcal{X} \subseteq \mathbb{R}^d$ be the data space, $\mathcal{Y} = \{1, 2, ..., K\}$ be the label space and $\mathcal{Y} \cup \{K+1\}$ be the open label space with $K+1$ as the class assigned to OOD data. We use $p_s(x, y = \cdot)$ and $p_t(x, y = \cdot)$ respectively to denote the source and target domain joint data and label distributions, and Δ^{K-1} to denote the K-dimensional probability simplex. To model ID versus OOD data, we introduce binary random variables (RVs) B_s, B_t in the source and target domains, respectively. $B_s, B_t = 1$ and $B_s, B_t = 0$ mean ID and OOD, respectively.

5.2.1 Definition and Assumptions

Similar to the Closed Set Label Shift problem (Definition 1), we study the Open Set Label Shift problem based on the assumption that source and target domain have identical conditional distributions of data x given label y (originally defined in Chapter 2, Assumption 2):

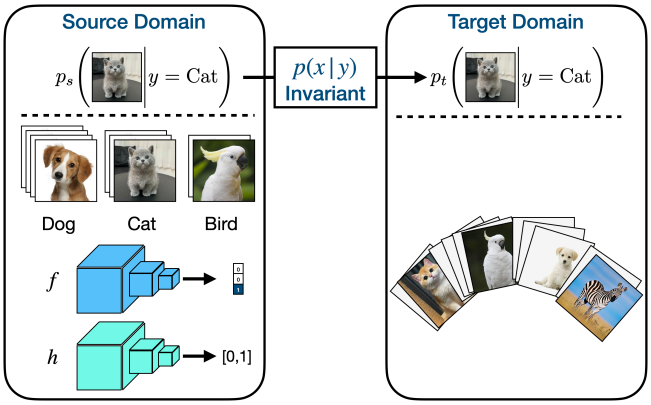

Figure 5.1: Open Set Label Shift problem setup, which includes: 1) **assumption**: Open Set Label Shift assumption (Assumption 2), 2) **datasets**: a source domain labeled dataset and a target domain unlabeled dataset (that includes an extra OOD class) and 3) **models**: a source domain classifier f for ID classes and a source domain ID/OOD binary classifier h. This information is used for the Open Set Label Shift ***detection***, **estimation** and **correction** problems.

Assumption 2. (Open Set Label Shift Assumption)

$$p_s(x|y=i) = p_t(x|y=i) \quad \text{for all} \quad i \in \mathcal{Y} \cup \{K+1\}, \tag{2.3}$$

where $K+1$ refers to the extra out-of-distribution class.

Recall that in the Open Set Label Shift problem, we can also define the *detection*, *estimation* and *correction* task:

Definition 2. (**Open Set Label Shift Problem**)

Under Assumption 2, given:

- Source domain ID labeled data $\mathcal{D}^s = \{(x_i^s, y_i^s)\}_{i=1}^{N^s}$ (K ID classes);
- Target domain unlabeled data $\mathcal{D}^t = \{x_i^t\}_{i=1}^{N^t}$ (K ID classes + 1 OOD class);
- Source domain ID classifier f;
- Source domain ID/OOD classifier h,

the Open Set Label Shift problem is to solve

- *Detection*: Verify $p_s(y) = p_t(y|y \in \mathcal{Y})$;
- *Estimation:* Estimate $p_t(y = \cdot)$;
- *Correction:* Model $p_t(y = \cdot|x)$ based on f and h.

A graphical model depiction of the OSLS estimation problem is given in Figure 5.2. We focus on the estimation and correction problems. Our overarching goal is to first estimate the target ID label distribution π and target ID data ratio ρ_t, then use these estimates to build a better classifier and OOD detector.

In Definition 2, the ID classifier f can be obtained by training a NN model on the source domain data via supervised learning. Any models proposed in the OOD detection literature can be used for the OOD classifier h. Due to the absence of

the OOD training data, h is usually established on intuitive principles (Hsu et al., 2020) and hence their test performance lacks theoretical guarantees. Our primary assumption is that regardless of their origins, both f and h can be understood as posterior predictive models, which respectively describe the probability of the ID label y given the input data x for ID data $b = 1$, and the probability that the data is in distribution given the input data x.

Assumption 5. For all $(x, y) \in \mathcal{X} \times (\mathcal{Y} \cup \{K+1\})$:

Assumption 5A $p_s(y|x, b = 1) = f(x)$, and
Assumption 5B $p_s(b = 1|x) = h(x)$.

We superimpose our assumptions onto the graphical model in Figure 5.2. The validity of Assumption 5A for practical classifiers is justified empirically in the experiments (§5.4.3) and the case when Assumption 5B is not satisfied is discussed in §5.3.4.

Practical Classifier Choices: In practice, an ID classifier that satisfies the OSLS problem setup (Definition 2) can be obtained by training a Neural Network classifier on the source domain dataset $\mathcal{D}^s$ through supervised learning. On the other hand, according to the OOD detection literature, the OOD classifier can be obtained based on the Neural Network ID classifier without ground truth OOD samples (Hsu et al., 2020; Liang et al., 2017). For example, OpenMax (Bendale and Boult, 2016) fit image features from each ID class with a Weibull distribution. The ID/OOD samples are distinguished based on the likelihood of the test samples for each ID class distribution. In this way, the ID/OOD classifier can be obtained with $\mathcal{D}^s$ given in Definition 2. All the ID/OOD classifiers we considered in the experiment satisfy this requirement.

5.2.2 Graphical model setup

Without loss of generality, we parameterize the source label distribution $Y_s|B_s = 1 \sim \text{Cat}(K, \boldsymbol{c})$ and target label distribution $Y_t|B_t = 1 \sim \text{Cat}(K, \boldsymbol{\pi})$ both as categorical distributions. Let the ID indicator B_s, B_t follow Bernoulli distributions $B_s \sim \text{Bern}(\rho_s)$ and $B_t \sim \text{Bern}(\rho_t)$, with $p_s(b = 1) = \rho_s$ and $p_t(b = 1) = \rho_t$ as the probability of the data being ID on source and target domain respectively. Formally, we are given:

$$
\begin{aligned}
p_s(y|b; \boldsymbol{c}) &= \begin{cases} c_j, & \text{if } b = 1, y \in \mathcal{Y} \\ 1, & \text{if } b = 0, y = K+1 \\ 0, & \text{otherwise} \end{cases}, \\
p_t(y|b; \boldsymbol{\pi}) &= \begin{cases} \pi_j, & \text{if } b = 1, y \in \mathcal{Y} \\ 1, & \text{if } b = 0, y = K+1 \\ 0, & \text{otherwise} \end{cases}.
\end{aligned}
\tag{5.1}
$$

The graphical model setup along with our assumptions are illustrated in Fig. 5.2. We optionally place priors over the target label ID/OOD ratio ρ_t and the target ID probabilities $\boldsymbol{\pi}$, which are further discussed in subsequent sections. We treat the source label ID/OOD ratio ρ_s and the source ID probabilities $\boldsymbol{c}$ as deterministic parameters.

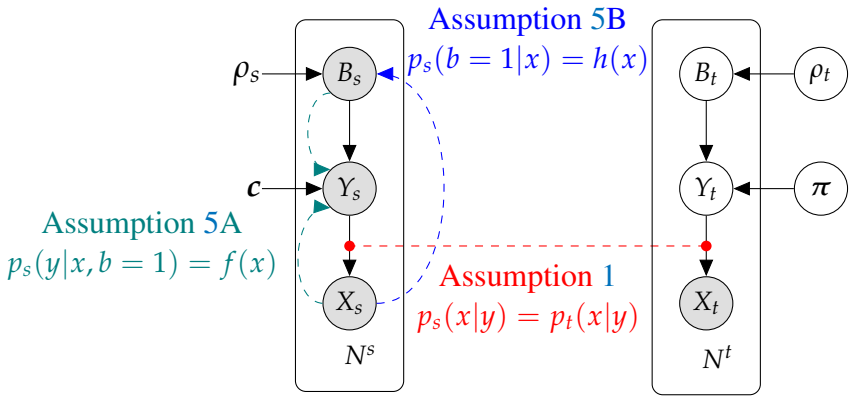

Figure 5.2: Graphical model of the Open Set Label Shift setting and our assumptions. X_s, X_t are data for the source and target domain, Y_s, Y_t are the corresponding categorical-valued labels and B_s, B_t are binary values representing ID/OOD data. $\boldsymbol{c}, \boldsymbol{\pi}$ are source and target domain label distribution class probabilities. The source domain random variable of the ID image $X_s|B_s=1$ and label $Y_s|B_s=1$ are observed with i.i.d. samples in $\mathcal{D}^s$ and the random variable of the OOD image $X_s|B_s=0$ is observed with i.i.d. samples in the reference dataset in $\mathcal{D}^{\mathbf{o}}$. The target domain random variable of the image X_t is observed with i.i.d. samples in $\mathcal{D}^t$. Under the Open Set Label Shift assumption (Assumption 2) and with the help of a reference OOD dataset $\mathcal{D}^{\text{o}}$ at test time, we first estimate $\rho_s, \boldsymbol{c}$ and then $\rho_t, \boldsymbol{\pi}$ without retraining. Optional prior distributions are employed on $\rho_t, \boldsymbol{\pi}$.

5.3 Proposed Method

5.3.1 Model Overview

Our proposed method solves the OSLS estimation and correction problem.

Method Summary: The idea of our approach is **reformulating the OSLS estimation problem as a Closed Set Label Shift (CSLS) estimation problem** and utilizing the Closed Set Label Shift methods to tackle the OSLS problem. The main challenge lies in both problem transformation and algorithm derivation.

Specifically, in the OSLS estimation problem, we reparameterize the likelihood of the target label distribution in the OSLS setting in a mathematical form similar to the likelihood in the CSLS setting in Section 5.3.3. This is achieved with the help of the estimate obtained in Section 5.3.2 with an OOD reference dataset. Then following the idea in the CSLS method (Chapter 3), an EM algorithm is then derived to obtain a MLE of the target label distribution for ID and OOD classes. Finally, a correction model is provided in Section 5.3.4 when some assumptions of the reparameterization are not satisfied. With the help of the proposed reparameterization, the label shift correction methods in the closed set setting can be directly used in the OSLS setting. A visualization of the model structure is provided in Fig. 5.3.

Remark on OOD Dataset $\mathcal{D}^{\mathbf{o}}$: Although our OSLS estimation model is derived by assuming $\mathcal{D}^{\mathbf{o}}$ contains ground truth OOD samples, this requirement can be relaxed (Section 5.3.5) so that pseudo OOD samples can be used instead and the OSLS problem setup (Definition 2) is satisfied. This use of pseudo OOD samples leads to our algorithms with demonstrated empirical performance benefits over existing

models, using the same form of input training data, as shown in the experiment section.

Model Outline: The estimation model includes:

1. (Section 5.3.2) The source ID data ratio ρ_s retrieval model to estimate ρ_s.
2. (Section 5.3.3) The EM algorithm based target label distribution estimation model to estimate π, ρ_t.
3. (Section 5.3.4) The target ID ratio ρ_t correction model to correct ρ_t.
4. (Section 5.3.5) Discussion on the choice of reference OOD dataset $\mathcal{D}^{\mathbf{o}}$ used in previous models.

An OSLS correction model is introduced in Section 5.3.6. Finally, the overall method is summarized in Section 5.3.7. All proofs can be found in Appendix. C.1.

5.3.2 Source ID/OOD Data Ratio retrieval

Several parameters have to be estimated before transforming the OSLS estimation problem into a CSLS estimation problem. We use the standard maximum likelihood estimator to estimate the source domain ID label distribution parameters $\boldsymbol{c} = p_s(y = \cdot)$, which amounts to computing the relative empirical frequencies of ID/OOD data on the source domain. However, estimating the probability of ID $p_s(b = 1) = \rho_s$ requires more careful attention.

This section aims to estimate the source domain label distribution $p_s(b = 1) = \rho_s$ (Fig. 5.2) under the OSLS problem setup, where only the source domain ID dataset $\mathcal{D}^s$ and an OOD classifier $h(x)$ is available. In this chapter, we treat $h(x)$ as a classifier pre-trained on some unknown source domain dataset with both ID and OOD data, and estimate ρ_s for that dataset with h, $\mathcal{D}^s$ and the reference OOD dataset $\mathcal{D}^{\mathbf{o}}$.

We consider the following estimate of ρ_s:

$$\hat{\rho}_s = \frac{\hat{\mu}_0}{1 - \hat{\mu}_1 + \hat{\mu}_0}, \tag{5.2}$$

where

$$\hat{\mu}_0 := \frac{1}{|\mathcal{D}^{\mathbf{o}}|} \sum_{x \in \mathcal{D}^{\mathbf{o}}} h(x) \quad \text{and} \quad \hat{\mu}_1 := \frac{1}{|\mathcal{D}^s|} \sum_{x \in \mathcal{D}^s} h(x). \tag{5.3}$$

Utilizing concentration inequalities (Vershynin, 2018) (Section 2.2.2), the error of the estimate may be quantified.

Theorem 8. ***(Source ID/OOD ratio estimator)*** *Under Assumption 5B, given source ID dataset $\mathcal{D}^s$ and source OOD dataset $\mathcal{D}^{\boldsymbol{o}}$, then for all $\delta > 0$, with probability of at least $1 - 2\delta$,*

$$|\rho_s - \hat{\rho}_s| \leq \frac{1}{1 - \mu_1 + \mu_0} \sqrt{\frac{\log 1/\delta}{2 \min(|\mathcal{D}^{\boldsymbol{o}}|, |\mathcal{D}^s|)}} \tag{5.4}$$

where

$$\mu_0 := \mathbb{E}_{X_s | B_s = 0}[h(x)] \quad \textit{and} \quad \mu_1 := \mathbb{E}_{X_s | B_s = 1}[h(x)]. \tag{5.5}$$

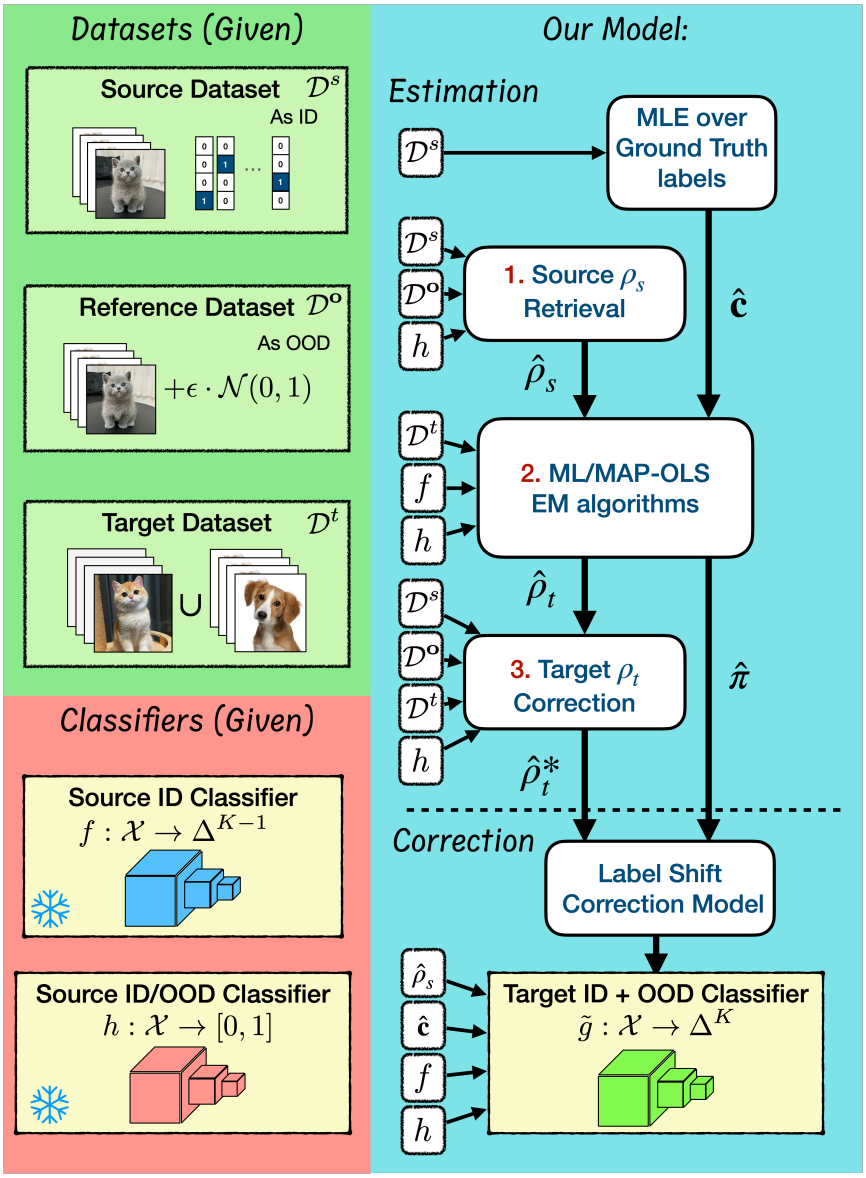

Figure 5.3: Structure of our proposed Open Set Label Shift estimation and correction method. The target ID label distribution probabilities $p_t(y = \cdot) = \boldsymbol{\pi}$ and ID data ratio $p_t(b = 1) = \rho_t$ are estimated through three steps: 1) retrieve source ID data ratio ρ_s (Section 5.3.2), 2) estimate target ID data ratio ρ_t and target ID label distribution $\boldsymbol{\pi}$ via an EM algorithm under Assumption 1,5 (Section 5.3.3) and 3) correct the target ID data ratio estimator $\hat{\rho}_t$ when OOD classifier $h(x)$ does not satisfy Assumption 5B (Section 5.3.4). Our correction model constructs a new classifier to classify target domain images based on the estimation result.

Here $|\mathcal{D}|$ returns the cardinality of the set $\mathcal{D}$. Detailed proof can be found in Appendix C.1.1.

Theorem 8 states that when the prerequisites are satisfied, the proposed estimator $\hat{\rho}_s$ is highly likely to be close to the ground truth ρ_s when the dataset $\mathcal{D}^s$ and $\mathcal{D}^{\mathbf{o}}$ have sufficient number of samples. Or more specifically, if Assumption 5B holds and the OOD dataset $\mathcal{D}^{\mathbf{o}}$ contains ground truth OOD samples, then for a small chosen $\delta > 0$, there is a large probability $1 - 2\delta$ that the proposed estimator $\hat{\rho}_s$ deviates from the ground ρ_s by small amount (right hand side of Eq. (5.4)), where such deviation decreases when more samples are given in the source ID dataset $\mathcal{D}^s$ and source OOD dataset $\mathcal{D}^{\mathbf{o}}$.

A variant of Theorem 8 can be shown and applied to the closed-set label shift, multi-class case (see Appendix C.1.2), which enables retrieval of the source label distribution $\boldsymbol{c}$ under the absence of the source domain dataset $\mathcal{D}^s$. We leave empirical investigation of the extended result for future work, as our work focuses on the OSLS problem.

5.3.3 EM algorithm for OSLS estimation

With the estimate of $\rho_s, \boldsymbol{c}$ obtained in the previous section, this section reformulates the OSLS estimation objective similar to a CSLS estimation objective and presents the EM algorithms to estimate the target label distribution $p_t(y = \cdot) = \boldsymbol{\pi}$ and target ID data ratio $p_t(b = 1) = \rho_t$.

Negative log likelihood Based on Assumptions A. 1 and A. 5, we can construct the negative log likelihood (NLL) of the target label distribution $p_t(y = \cdot) = \boldsymbol{\pi}$ and ID data ratio $p_t(b = 1) = \rho_t$ given target unlabeled data $\mathcal{D}^t$.

Lemma 9. *Under Assumption 1 and Assumption 5, given $\mathcal{D}^t$, the negative log likelihood of parameter $\boldsymbol{\pi}$ and ρ_t can be written as:*

$$-\log L(\boldsymbol{\pi}, \rho_t; \mathcal{D}^t) = -\sum_{i=1}^{N^t} \log \left(\sum_{j=1}^{K+1} \frac{\tilde{\pi}_j}{\tilde{c}_j} \tilde{f}(x_i)_j \right) + C, \tag{5.6}$$

where C does not depend on either $\boldsymbol{\pi}$ or ρ_t and

$$\tilde{f}(x)_i := \begin{cases} h(x) \cdot f(x)_i, & i \in \mathcal{Y} \\ 1 - h(x), & i = K + 1, \end{cases} \tag{5.7}$$

$$\begin{aligned} \tilde{\boldsymbol{\pi}} &:= [\rho_t \cdot \pi_1, ..., \rho_t \cdot \pi_K, 1 - \rho_t]^T \\ \tilde{\boldsymbol{c}} &:= [\rho_s \cdot c_1, ..., \rho_s \cdot c_K, 1 - \rho_s]^T. \end{aligned} \tag{5.8}$$

The NLL in Eq. (5.6) is mathematically similar to the NLL analyzed in the MLLS model (Saerens et al., 2002), which is proposed for the CSLS estimation problem (Section 2.3.1). The detailed derivation can be found in Appendix. C.1.3.

Maximum Likelihood Estimate That the negative log likelihood in Eq. (5.6) has the same form of the NLL of a Closed Set Label Shift estimation problem with $K + 1$

classes (see MLLS model in Chapter 2, Section 2.3.1 or Alexandari et al. (2020)). Observing this similarity, we can minimize the NLL in Eq. (5.6) by viewing $\tilde{f}$ as the closed set $K+1$ class classifier and $\tilde{\boldsymbol{c}}, \tilde{\boldsymbol{\pi}}$ as parameters of the closed set source and target label distribution. The MLE of target label distribution parameters $\boldsymbol{\pi}^{\text{MLE}}, \rho_t^{\text{MLE}}$ can be obtained via:

$$\boldsymbol{\pi}^{\text{MLE}}, \rho_t^{\text{MLE}} \in \underset{\tilde{\pi} \in \Delta^K}{\arg\min} - \log L(\boldsymbol{\pi}, \rho_t; \mathcal{D}^t). \tag{5.9}$$

Although the MLE objective in Eq. (5.9) is not convex in $(\boldsymbol{\pi}, \rho_t)$, it is convex in $\tilde{\boldsymbol{\pi}}$ – a reparameterization of the parameter $\boldsymbol{\pi}$ and ρ_t. Therefore we may still derive an EM-algorithm that converges to the global minimum. Inspired by Saerens et al. (2002), we compute a reparameterized MLE of $\tilde{\boldsymbol{\pi}}$ in Eq. (5.8). As MLE is invariant under reparameterization (Murphy, 2012) (Chapter 2, Section 2.2.2), the MLE of $\tilde{\boldsymbol{\pi}}$ can be mapped back to the MLE of $\boldsymbol{\pi}$ and ρ_t. Details are described in Theorem 10 with proof available in Appendix C.1.4.

Theorem 10. ***(MLE)*** *Under Assumption 1 and Assumption 5, the the NLL (5.6) is convex in $\tilde{\boldsymbol{\pi}}$ (and convex in ρ_t), and the EM algorithm MLE-OLS (Alg. 7) converges to $\boldsymbol{\pi}^{MLE}, \rho_t^{MLE}$ (5.9).*

Maximum *a Posteriori* **estimate** We may also attempt to compute MAP estimates $\boldsymbol{\pi}^{\text{MAP}}, \rho_t^{\text{MAP}}$, when prior information about the two parameters are available. However the MAP is not invariant under reparameterisations, and the posterior probability density is nonconvex in $(\boldsymbol{\pi}, \rho_t)$. This makes it challenging to compute MAP estimates, and in this sense, the MLE is favourable. We detail the use of a Dirichlet prior over $\boldsymbol{\pi} \sim \text{Dir}(K, \boldsymbol{\alpha}^{\textbf{in}})$ and a Beta prior over $\rho_t \sim \text{Beta}(\alpha_1^{\textbf{out}}, \alpha_2^{\textbf{out}})$ in Appendix C.1.5.

Alg. 7 summarises the EM algorithm for MLE and MAP estimation ($\mathbb{R}^K_{>1} := \{x \in \mathbb{R}^K | x_i > 1, i = 1, ..., K\}$).

5.3.4 Target ID/OOD Data Ratio Correction

In Assumption 5, we describe the conditional distribution $p(b = 1|x)$ with an OOD classifier $h(x)$. In practice, however, the OOD classifiers can yield unsatisfactory performance due to the challenging OOD detection problem setup (Zhang, Yang et al., 2023). Deploying such a classifier in the OSLS algorithms can result in high estimation error.

This section provides a correction model to mitigate the possible estimation error on ρ_t. We find that ρ_t can still be estimated with a practical OOD classifier h' that does not satisfy Assumption 5B, if $h'(x)$ has different expected responses to ID and OOD samples but identical response to samples in different ID classes:

$$\begin{aligned} &\mathbb{E}_{X_s|Y_s=i}[h'(x)] \neq \mathbb{E}_{X_s|Y_s=K+1}[h'(x)] \quad \text{and} \\ &\mathbb{E}_{X_s|Y_s=i}[h'(x)] = \mathbb{E}_{X_s|Y_s=j}[h'(x)] \text{ for all } i, j \in \mathcal{Y}. \end{aligned} \tag{5.12}$$

Algorithm 7 MLE/MAP-OLS

Input: $\mathcal{D}^t = \{x_i^t\}_{i=1}^{N^t}, \boldsymbol{c}, \rho_s, h(x), f(x)$,
- MLE-OLS: $\boldsymbol{\alpha}^{\mathbf{in}} = \mathbf{1}, \alpha_1^{\mathbf{out}}, \alpha_2^{\mathbf{out}} = \mathbf{1}$.
- MAP-OLS: $\boldsymbol{\alpha}^{\mathbf{in}} \in \mathbb{R}_{>1}^K, \alpha_1^{\mathbf{out}}, \alpha_2^{\mathbf{out}} \in \mathbb{R}_{>1}$.

Initialize: $\boldsymbol{\pi}^{(0)} \in \Delta_{>0}^{K-1}, \rho_t^{(0)} \in (0,1)$.
Construct: $\tilde{f}, \tilde{c}$ based on Eq. (5.7), Eq. (5.8).
for $m = 0$ to M **do**
Construct: $\tilde{\boldsymbol{\pi}}^{(m)}$ based on $\boldsymbol{\pi}^{(m)}, \rho_t^{(m)}$ and Eq. (5.8).
E-step: For $j \in \mathcal{Y} \cup \{K+1\}$, evaluate

$$g_{ij}^{(m)} = \frac{\tilde{\pi}_j^{(m)} / \tilde{c}_j \cdot \tilde{f}(x_i^t)_j}{\sum_{l=1}^K \tilde{\pi}_l^{(m)} / \tilde{c}_l \cdot \tilde{f}(x_i^t)_l}. \tag{5.10}$$

M-step: For $j \in \mathcal{Y}$, evaluate

$$\begin{cases} \pi_j^{(m+1)} = \dfrac{\sum_{i=1}^{N^t} g_{ij}^{(m)} + \alpha_j^{\mathbf{in}} - 1}{N^t - \sum_{i=1}^{N^t} g_{iK+1}^{(m)} + \sum_{l=1}^K (\alpha_l^{\mathbf{in}} - 1)} \\ \rho_t^{(m+1)} = \dfrac{N^t - \sum_{i=1}^{N^t} g_{iK+1}^{(m)} + \alpha_1^{\mathbf{out}} - 1}{N^t + \alpha_1^{\mathbf{out}} + \alpha_2^{\mathbf{out}} - 2}. \end{cases} \tag{5.11}$$

end for
Output: $p_t(y = \cdot) = \boldsymbol{\pi}^{(M+1)}, p_t(b = 1) = \rho_t^{(M+1)}$.

Theorem 11. *(**Target ID/OOD ratio correction**) Under Assumption 1 and Assumption 5A (without 5B), for a classifier $h' : \mathcal{X} \to [0,1]$ that satisfies Eq. (5.12), given source ID dataset $\mathcal{D}^s$, OOD dataset $\mathcal{D}^o$, target dataset $\mathcal{D}^t$, then for $\delta > 0$, with probability of at least $1 - 2\delta$ we have:*

$$|\rho_t - \hat{\rho}_t^*| \leq \frac{1}{|\mu_1' - \mu_0'|} \sqrt{\frac{2 \log 1/\delta}{\min(|\mathcal{D}^s|, |\mathcal{D}^o|, |\mathcal{D}^t|)}}, \tag{5.13}$$

where

$$\hat{\rho}_t^* = \frac{\hat{\rho}' - \hat{\mu}_0'}{\hat{\mu}_1' - \hat{\mu}_0'}, \quad \textit{and} \quad \hat{\rho}' := \frac{1}{|\mathcal{D}^t|} \sum_{x_i \in \mathcal{D}^t} h'(x_i), \tag{5.14}$$

with

$$\hat{\mu}_0' := \frac{1}{|\mathcal{D}^o|} \sum_{x \in \mathcal{D}^o} h'(x), \qquad \hat{\mu}_1' := \frac{1}{|\mathcal{D}^s|} \sum_{x \in \mathcal{D}^s} h'(x), \tag{5.15}$$

and

$$\mu_0' := \mathbb{E}_{X_s | B_s = 0}[h'(x)], \qquad \mu_1' := \mathbb{E}_{X_s | B_s = 1}[h'(x)]. \tag{5.16}$$

Detailed proof can be found in Appendix C.1.6.

Theorem 11 states that when the prerequisites are satisfied, the proposed estimator $\hat{\rho}_t^*$ is highly likely to be close to the ground truth ρ_t when the dataset $\mathcal{D}^s$, $\mathcal{D}^{\mathbf{o}}$ and $\mathcal{D}^t$ have sufficient number of samples. Or more specifically, under the required conditions, for

a small chosen $\delta > 0$, there is a large probability $1 - 2\delta$ that the proposed estimator $\hat{\rho}_t^*$ deviates from the ground ρ_t by a small amount (right hand side of Eq. (5.13)), where such deviation decreases when more samples are given in the source ID dataset $\mathcal{D}^s$, the source OOD dataset $\mathcal{D}^{\mathbf{o}}$ and the target unlabeled dataset $\mathcal{D}^t$.

The condition Eq. (5.12) in Theorem 11 is a reasonable assumption because the first equation holds when h' can roughly separate ID/OOD samples in the output space $[0,1]$. The second equation holds when h' is trained/constructed based on a class-uniform ID dataset.

Based on Eq. (5.14) in Theorem 11, we propose a correction model for the ρ_t^{MLE} and ρ_t^{MAP} obtained in Alg. 7 via:

$$\rho_t^{\text{MLE}*} = \frac{\rho_t^{\text{MLE}} - \hat{\mu}_0}{\hat{\mu}_1 - \hat{\mu}_0} \quad \text{and} \quad \rho_t^{\text{MAP}*} = \frac{\rho_t^{\text{MAP}} - \hat{\mu}_0}{\hat{\mu}_1 - \hat{\mu}_0}. \tag{5.17}$$

Further discussion about Eq. (5.17) are provided in Appendix C.1.7 and empirical analysis in Fig. 5.4.

5.3.5 Choice of OOD Reference Dataset

In our OSLS estimation model, only $\hat{\mu}_0$ in our OSLS model directly depends on the OOD reference dataset $\mathcal{D}^{\mathbf{o}}$ (Eq. (5.3) and Eq. (5.17)). Thus as long as the expectation of $h(x)$ on the distribution that generates $\mathcal{D}^{\mathbf{o}}$ equals to the expectation of $h(x)$ on the ground truth OOD distribution, our model can yield desired estimates. In this Chapter, we generate the OOD reference dataset by a linear combination of Gaussian noise and the source domain ID samples in $\mathcal{D}^s$. With $\gamma \in (0,1)$ we have:

$$\mathcal{D}_\gamma^{\mathbf{o}} = \{(1-\gamma)\cdot x_i + \gamma \cdot \epsilon | x_i \in \mathcal{D}^s, \epsilon \sim \mathcal{N}(0,1)\}, \tag{5.18}$$

We choose γ to be close to 0 so that samples of $\mathcal{D}_\gamma^{\mathbf{o}}$ will be close to the ground truth ID samples in $\mathcal{D}^s$. In this case, the $\hat{\mu}_0$ the computed by $\mathcal{D}_\gamma^{\mathbf{o}}$ could be higher than that obtained with actual OOD samples, therefore we introduce another re-weight factor T so that:

$$\hat{\mu}_0^* = \frac{1}{|\mathcal{D}_\gamma^{\mathbf{o}}|T} \sum_{x_i \in \mathcal{D}_\gamma^{\mathbf{o}}} h(x_i), \tag{5.19}$$

which is then used as $\hat{\mu}_0$ in Eq. (5.3) and Eq. (5.17) in our model.

5.3.6 OSLS correction method

The OSLS correction model can be implemented based on the Closed Set Label Shift correction model of $K+1$ classes (Ye et al., 2024). With estimates of the parameters $\boldsymbol{c}, \rho_s$ of the source label distribution and parameters $\boldsymbol{\pi}, \rho_t$ of the target label distribution, we can construct the source and target label distribution of all classes with $\tilde{c}, \tilde{\pi}$ based on Eq. (5.8) and the source domain classifier $\tilde{f}$ in Eq. (5.7) for

$K+1$ classes. The target domain classifier can be constructed via:

$$\tilde{g}(x) = \frac{\frac{\tilde{\pi}_j}{\tilde{c}_j}\tilde{f}(x)_j}{\sum_{l=1}^{K+1}\frac{\tilde{\pi}_l}{\tilde{c}_l}\tilde{f}(x)_l}. \tag{5.20}$$

5.3.7 Overall Framework

Our practical OSLS-EM model estimate and correction Open Set Label Shift follows the procedure in Alg. 8.

Algorithm 8 OSLS-MLE/MAP Framework

Input: $\mathcal{D}^t, \mathcal{D}^s, h(x), f(x)$, hyper-params γ, T.
Optional: (MAP prior) $\boldsymbol{\alpha}^{\text{in}} \in \mathbb{R}^K_{>1}$, $\alpha_1^{\text{out}}, \alpha_2^{\text{out}} \in \mathbb{R}_{>1}$.
OOD Dataset: Generate OOD dataset $\mathcal{D}^{\mathbf{o}}$ with Eq. (5.18).
Estimate $\boldsymbol{c}$ with ground truth labels in $\mathcal{D}^s$.
OSLS estimation:
1. **Source** ρ_s **Retrieval**: Obtain $\hat{\rho}_s$ in Eq. (5.2) with $\hat{\mu}_0$ in Eq. (5.3), $\hat{\mu}_0^*$ in Eq. (5.19).
2. **EM algorithm Estimation**: Obtain $\hat{\rho}_t, \hat{\boldsymbol{\pi}}$ in terms of MLE/MAP with Alg. 7.
3. **Target** ρ_t **Correction**: Obtain corresponding $\hat{\rho}_t^*$ via Eq. (5.14).

OSLS correction: Obtain $g(x)$ with Eq. (5.20) based on the estimates $\hat{\rho}_t^*, \hat{\boldsymbol{\pi}}$.

5.4 Experiments

5.4.1 Experimental Setup

Datasets: Following the experimental setup in the OOD detection literature (Zhang, Yang et al., 2023), we evaluate our model with CIFAR10, CIFAR100 (Krizhevsky, Hinton et al., 2009) and ImageNet (Russakovsky et al., 2015a) dataset as ID datasets and with SVHN (Netzer et al., 2011), Places (Zhou, Lapedriza et al., 2017), OpenImage-O (Wang, Li et al., 2022b), NINCO (Bitterwolf et al., 2023), a subset of TinyImageNet (Krizhevsky, Sutskever et al., 2012), a subset of iNaturalist (Huang and Li, 2021), a subset of Species (SSB) (Hendrycks, Basart et al., 2019b) datasets as OOD datasets. The OOD datasets are split into near OOD and far OOD groups depending on their similarity to the ID dataset. Details of the dataset setup are provided in Tab. 5.1, and further details are available in Appendix. C.2.

Our model is tested with different types of label shift, including the Dirichlet shift and the Ordered Long-Tailed (LT) shift commonly used in Closed Set Label Shift literature (Alexandari et al., 2020; Lipton et al., 2018). The Dirichlet shift adjusts ground truth $\boldsymbol{\pi}$ by sampling from a Dirichlet distribution with parameter α and the Ordered LT shift adjusts $\boldsymbol{\pi}$ based on a Long-Tailed distribution with different imbalance factor and “Forward" or “Backward" order (Ye et al., 2024). Under the open set setting, we also sub-sample the OOD datasets so that the test datasets have different OOD over ID ratios ($r = (1-\rho_t)/\rho_t$). Details can be found in Tab. 5.2.

ID dataset	OOD dataset	
CIFAR10	Near	CIFAR100, TinyImageNet
	Far	MNIST, SVHN, Texture, Places,
CIFAR100	Near	CIFAR10, TinyImageNet,
	Far	MNIST, SVHN, Texture, Places
ImageNet-200	Near	SSB, NINCO,
	Far	iNaturalist, Texture, OpenImage-O

Table 5.1: Dataset setup in our Open Set Label Shift experiment. For each ID dataset, different OOD datasets are tested to justify the performance of our OSLS estimation and correction model.

Label Shift	Shift Parameters	OOD/ID data ratio r
Dirichlet	$\alpha = 1.0, 10.0$	$r = 1, 0.5, 0.1, 0.01$
Ordered LT	$100, 50, 20, 10, 5$ "Forward/Backward"	$r = 1, 0.5, 0.1, 0.01$

Table 5.2: Types of label shift in our experiment, including Dirichlet shift with different shift parameter α and Ordered Long-Tailed (LT) shift with different imbalance factors under forward and backward order.

Model Setup: The Neural Network ID classifiers are implemented using PyTorch (Paszke et al., 2017). Following the convention in the OOD literature (Zhang, Yang et al., 2023), we train a ResNet18 (He, Zhang et al., 2016) on CIFAR10/100 and ImageNet-200 datasets as multi-class ID classifiers. We test our model with different OOD classifiers, including OpenMax (Bendale and Boult, 2016), Ash (Djurisic et al., 2022), MLS (Hendrycks, Basart et al., 2019a), ReAct (Sun, Guo et al., 2021) and KNN (Sun, Ming et al., 2022), with the implementations provided by the OpenOOD project (Zhang, Yang et al., 2023). The outputs of these classifiers are re-scaled to $[0, 1]$ range to satisfy the requirement of our model $h : \mathcal{X} \to [0, 1]$.

In the estimation model, we follow the MAPLS (Ye et al., 2024) setup (soft mode) to initialize the EM algorithms MLE/MAP-OLS with $\boldsymbol{\pi} = \boldsymbol{c}$ and $\rho_s = \rho_t$ and run for 100 iterations to ensure convergence. More details are given in Appendix. C.2.

Evaluation Metrics: We evaluate our model mainly on the label shift estimation error $(w - \hat{w})^2 / K$ (Lipton et al., 2018) over ID classes and the Top1 accuracy over all ID and OOD classes. The label shift estimation error is the MSE between the ground truth target over source ID label distribution ratio $w = \boldsymbol{\pi} / \boldsymbol{c}$ and $\hat{w}$ is the one that was obtained with the estimator of $\boldsymbol{\pi}$. The comparison of ground truth ID data ratio ρ_t and estimate $\hat{\rho}_t^*$ are also compared.

5.4.2 Results Comparison

We report the performance of our OSLS-MAP model. As the Open Set Label Shift problem has been studied only recently, we mainly compare the performance of our model with state-of-the-art (SOTA) Closed Set Label Shift estimation models MLLS (Saerens et al., 2002), BBSE (Lipton et al., 2018), RLLS (Azizzadenesheli et al., 2018), MAPLS (Ye et al., 2024), and a baseline model that directly assumes

the target domain has uniform ID label distribution $\boldsymbol{\pi} = \mathbf{1}/K$ and same amount of ID/OOD data ($r = 1$). The baseline model considers the situation when no OSLS estimation model is available. In this case, it is natural to assume the target domain has an uniform ID label distribution $\boldsymbol{\pi} = \mathbf{1}/K$ (used in the Closed Set Label Shift model (Ye et al., 2024)) and same amount of ID/OOD data $r = 1$ (used in the OOD detection model (Meinke, Bitterwolf et al., 2022)).

For accuracy metric, OpenMax also proposed the ID + OOD classifier, which is reported as the "Original" model in Tab. 5.3. The model proposed by (Garg, Balakrishnan et al., 2022) is not compared because they adopt a domain adaptation approach and require retraining the OOD and ID classifier for each experiment setup, which is time-consuming, especially in large-scale datasets like ImageNet-200. Further, they have not reported their performance on the estimation error $(w - \hat{w})^2/K$.

Top1 Accuracy: As shown in Tab. 5.3, our model consistently outperforms baselines on most settings based on different OOD classifiers. Among the OOD classifiers that we tested, OpenMax exhibits better overall accuracy. This is probably because the OpenMax OOD classifier is designed to output $[0, 1]$ confidence and may better satisfy Assumption 5B for our model. Other OOD models usually output real number (*e.g.* max logit) that requires re-scaling, where the re-scale process can introduce extra estimation error.

Estimation Error: As seen in Tab. 5.4, 5.5 and 5.6, our estimation model effectively estimates ID label shift in the open set settings on CIFAR10/100 and ImageNet-200 datasets and outperforms the open set baseline in most settings. Moreover, although the performance of the closed set models increases when the target domain has fewer OOD samples (small r), our model takes OOD data into account and still outperforms all existing SOTA closed set models in the reported Open set settings. Similar to the Top1 Accuracy result, in terms of the estimation error, OpenMax fits better with our Assumption 5B and thus also performs the best among the OOD classifiers in most of the OSLS settings in the table.

ρ_t Correction Results: Fig. 5.4 justifies our ρ_t correction model (Section 5.3.4) with results on CIFAR10/100 dataset under Dirichlet ID shift. As seen in the figure, most estimates $\hat{\rho}_t^*$ of our model match better with the ground truth ρ_t than $\hat{\rho}_t$ obtained without our ρ_t correction model. Such a result implies that the tested OOD classifiers roughly satisfy the requirement of Theorem 11 in Eq. (5.12). This is probably because these OOD classifiers are usually designed based on the maximal output of the ID classifier (*e.g.* max logit), and such output tends to be identical among ID classes when the source domain ID dataset that the ID classifier trained on is class-uniform.

Estimation Error on Imbalanced Train set: Similar to the closed set label shift models, we also study the robustness of the open set label shift model under imbalanced source label distributions. We report the estimation error performance of our OSLS-MAP model on the Long-Tailed CIFAR10 dataset in Tab. 5.7, 5.8 and on the Long-Tailed CIFAR100 dataset in Tab. 5.9, 5.10. As shown in the tables, our model exhibit decent performance in most the "Forward" cases, where the target label distribution is closed to the Long-Tailed source label distribution. When the source and target domain have very different label distributions in the "Backward" cases,

our model is still outperform the closed set label shift models in most of the settings, especially in the extreme LT50/LT100 case.

Dataset			CIFAR10						CIFAR100					
ID label Shift param			LT10		LT50		LT100		LT10		LT50		LT100	
OOD label shift param r			1.0	0.1	1.0	0.1	1.0	0.1	1.0	0.1	1.0	0.1	1.0	0.1
OpenMax	Near	Original	74.65	86.80	74.34	87.65	74.75	87.16	63.54	65.20	63.70	65.12	63.42	64.86
		Baseline	75.33	86.58	75.06	87.34	75.30	86.93	62.71	64.92	62.95	64.72	62.67	64.61
		ours	75.33	86.58	75.06	87.39	75.30	86.98	62.95	65.25	63.43	65.78	63.23	66.03
	Far	Original	76.56	87.11	76.55	87.94	76.73	87.53	66.77	65.93	67.11	65.95	66.76	65.32
		Baseline	77.39	86.91	77.40	87.70	77.57	87.33	66.58	65.73	66.99	65.69	66.58	65.19
		ours	77.39	86.91	77.41	87.75	77.61	87.41	66.78	66.03	67.37	66.73	67.24	66.67
MLS	Near	Baseline	80.92	79.44	80.47	80.35	80.90	79.85	71.22	57.55	70.92	56.74	70.56	57.12
		ours	80.92	79.44	80.47	80.36	80.90	79.90	71.25	57.70	71.09	57.15	70.81	57.56
	Far	Baseline	83.90	80.04	83.58	80.81	84.02	80.26	70.77	57.49	70.34	56.66	70.11	56.93
		ours	83.90	80.04	83.59	80.82	84.04	80.31	70.81	57.63	70.54	57.06	70.40	57.38
ReAct	Near	Baseline	80.12	78.45	79.33	77.72	79.17	77.04	70.78	57.31	70.83	56.46	70.64	57.22
		ours	80.12	78.46	79.32	77.72	79.17	77.04	70.81	57.52	70.99	56.87	70.88	57.65
	Far	Baseline	82.62	79.04	82.12	78.02	81.57	77.61	71.15	57.17	71.11	56.70	70.77	57.29
		ours	82.62	79.05	82.12	78.02	81.57	77.61	71.16	57.41	71.32	57.09	71.07	57.73
KNN	Near	Baseline	80.76	84.74	80.97	84.64	81.12	85.03	71.11	59.24	70.50	59.41	70.72	59.07
		ours	80.76	84.74	80.97	84.67	81.12	85.10	71.17	59.42	70.72	59.86	70.92	59.63
	Far	Baseline	84.03	85.23	84.16	85.46	84.46	85.65	72.22	59.47	71.82	59.89	71.76	59.11
		ours	84.03	85.23	84.18	85.49	84.49	85.74	72.28	59.64	72.04	60.34	71.97	59.69
Ash	Near	Baseline	68.48	68.57	68.19	68.65	67.95	68.77	68.32	57.00	67.77	56.04	67.89	56.30
		ours	68.49	68.58	68.21	68.66	67.96	68.84	68.37	57.09	67.95	56.40	68.11	56.84
	Far	Baseline	70.90	68.98	70.77	69.10	70.33	69.41	70.01	57.24	69.92	56.53	69.77	56.67
		ours	70.91	69.00	70.78	69.11	70.36	69.50	70.05	57.31	70.10	56.93	70.05	57.21

Table 5.3: Top1 Accuracy (ID + OOD) of our model on CIFAR10 and CIFAR100 dataset with OOD datasets (Near & Far) comparison under different ID and OOD label shift. Outperforming results are colored in gray. Our model outperforms baselines and other models under most label shift settings. Each metric is averaged among the corresponding OOD test set (Tab. 5.1) and over three independent ID classifiers.

Dataset			CIFAR10								
ID label Shift param			LT10 Forward			LT50 Forward			LT100 Forward		
OOD label shift param r			1.0	0.1	0.01	1.0	0.1	0.01	1.0	0.1	0.01
Closed Set Label Shift estimation models											
BBSE		Near	0.172	0.005	0.003	0.341	0.008	0.007	0.418	0.011	0.010
		Far	0.400	0.014	0.004	0.547	0.016	0.008	0.629	0.018	0.010
MLLS		Near	0.168	0.005	0.002	0.318	0.007	0.002	0.388	0.010	0.002
		Far	0.458	0.012	0.002	0.587	0.013	0.002	0.662	0.015	0.002
RLLS		Near	0.486	0.486	0.486	1.199	1.198	1.198	1.538	1.537	1.536
		Far	0.486	0.486	0.486	1.199	1.198	1.198	1.538	1.537	1.536
MAPLS		Near	0.177	0.015	0.007	0.369	0.034	0.014	0.458	0.043	0.017
		Far	0.396	0.019	0.007	0.565	0.036	0.014	0.659	0.045	0.017
Open Set Label Shift estimation models											
Baseline			0.487	0.487	0.487	1.200	1.200	1.200	1.540	1.540	1.540
ours	OpenMax	Near	**0.068**	**0.002**	0.002	**0.131**	**0.003**	0.002	**0.165**	**0.003**	**0.002**
		Far	**0.186**	**0.003**	0.002	**0.244**	**0.005**	**0.002**	**0.264**	**0.004**	**0.002**
	MLS	Near	**0.026**	0.007	0.007	**0.056**	0.007	0.010	**0.067**	**0.009**	0.008
		Far	**0.042**	**0.007**	0.007	**0.057**	**0.008**	0.010	**0.061**	**0.009**	0.008
	ReAct	Near	**0.060**	0.017	0.017	**0.098**	0.026	0.023	**0.114**	0.029	0.028
		Far	**0.084**	0.017	0.017	**0.110**	0.026	0.023	**0.119**	0.029	0.028
	KNN	Near	**0.035**	0.006	0.006	**0.071**	**0.005**	0.008	**0.090**	**0.006**	0.005
		Far	**0.075**	**0.006**	0.006	**0.090**	**0.006**	0.008	**0.104**	**0.006**	0.005
	Ash	Near	0.280	0.221	0.211	0.412	0.319	0.332	0.452	0.359	0.366
		Far	**0.393**	0.225	0.211	**0.521**	0.322	0.333	**0.568**	0.364	0.367

Table 5.4: Estimation Error $(w - \hat{w})^2/K(\downarrow)$ of our OSLS estimation model on CIFAR10 dataset with Near OOD datasets and Far OOD datasets comparison under Ordered-LT (Forward) ID and OOD label shift. Settings in which our model outperforms baselines are colored in gray. Our model outperforms baselines under most label shift settings. Each metric is averaged among corresponding OOD test set (Tab. 5.1) and over three independent ID classifiers.

Dataset			CIFAR100								
ID label Shift param			LT10 Forward			LT50 Forward			LT100 Forward		
OOD label shift param r			1.0	0.1	0.01	1.0	0.1	0.01	1.0	0.1	0.01
Closed Set Label Shift estimation models											
BBSE		Near	0.529	0.131	0.097	0.750	0.173	0.176	0.850	0.173	0.167
		Far	4.118	0.250	0.099	4.362	0.290	0.176	4.489	0.294	0.168
MLLS		Near	0.870	0.116	0.080	1.005	0.142	0.116	1.100	0.132	0.113
		Far	9.656	0.328	0.083	9.804	0.364	0.119	9.862	0.353	0.117
RLLS		Near	0.426	0.425	0.425	1.100	1.099	1.099	1.404	1.402	1.402
		Far	0.426	0.425	0.425	1.100	1.099	1.099	1.404	1.403	1.402
MAPLS		Near	0.672	0.116	0.085	0.860	0.161	0.129	0.965	0.164	0.134
		Far	7.481	0.275	0.087	7.667	0.330	0.131	7.763	0.336	0.137
Open Set Label Shift estimation models											
Baseline			0.426	0.426	0.426	1.101	1.101	1.101	1.405	1.405	1.405
ours	OpenMax	Near	**0.387**	**0.043**	**0.046**	**0.450**	**0.071**	**0.071**	**0.511**	**0.081**	**0.077**
		Far	2.223	**0.087**	**0.046**	2.353	**0.111**	**0.073**	2.341	**0.118**	**0.078**
	MLS	Near	**0.323**	**0.078**	**0.078**	**0.371**	**0.120**	0.121	**0.415**	**0.126**	**0.120**
		Far	1.289	**0.099**	**0.079**	1.324	**0.138**	0.123	**1.366**	**0.150**	**0.120**
	ReAct	Near	**0.331**	**0.076**	**0.075**	**0.362**	**0.114**	0.116	**0.396**	**0.110**	0.124
		Far	1.138	**0.096**	**0.075**	1.199	**0.131**	0.117	**1.202**	**0.127**	0.124
	KNN	Near	0.736	0.141	0.139	0.805	0.206	0.205	**0.817**	0.235	0.228
		Far	1.188	**0.152**	0.140	1.281	**0.216**	0.207	**1.287**	**0.246**	0.229
	Ash	Near	**0.358**	**0.111**	0.101	**0.514**	0.194	0.176	**0.541**	0.217	0.198
		Far	1.015	**0.119**	0.101	1.169	**0.206**	0.175	**1.183**	**0.229**	0.198

Table 5.5: Estimation Error $(w - \hat{w})^2/K(\downarrow)$ of our OSLS estimation model on CIFAR100 dataset with Near OOD datasets and Far OOD datasets comparison under Ordered-LT (Forward) ID and OOD label shift. Settings in which our model outperforms baselines are colored in gray. Our model outperforms baselines under most label shift settings. Each metric is averaged among corresponding OOD test set (Tab. 5.1) and over three independent ID classifiers.

Dataset			ImageNet-200								
ID label Shift param			LT10 Forward			LT50 Forward			LT100 Forward		
OOD label shift param r			1.0	0.1	0.01	1.0	0.1	0.01	1.0	0.1	0.01
Closed Set Label Shift estimation models											
BBSE		Near	0.564	0.119	0.107	0.664	0.128	0.115	0.735	0.132	0.112
		Far	1.148	0.134	0.108	1.301	0.141	0.115	1.389	0.146	0.112
MLLS		Near	1.152	0.131	0.099	1.233	0.143	0.111	1.272	0.146	0.116
		Far	4.095	0.167	0.101	4.270	0.182	0.112	4.436	0.189	0.117
RLLS		Near	0.432	0.432	0.432	1.083	1.082	1.082	1.397	1.396	1.396
		Far	0.433	0.432	0.432	1.083	1.082	1.082	1.397	1.396	1.396
MAPLS		Near	0.877	0.114	0.085	0.986	0.128	0.093	1.046	0.134	0.095
		Far	3.004	0.139	0.086	3.177	0.156	0.094	3.328	0.164	0.097
Open Set Label Shift estimation models											
Baseline			0.436	0.436	0.436	1.090	1.090	1.090	1.405	1.405	1.405
ours	OpenMax	Near	0.699	**0.035**	**0.022**	0.769	**0.035**	**0.020**	0.820	**0.035**	**0.019**
		Far	2.500	**0.048**	**0.022**	2.652	**0.054**	**0.020**	2.739	**0.047**	**0.019**
	MLS	Near	**0.194**	**0.069**	**0.069**	**0.209**	**0.078**	**0.076**	**0.217**	**0.079**	**0.081**
		Far	**0.118**	**0.069**	**0.069**	**0.128**	**0.078**	**0.076**	**0.126**	**0.081**	**0.082**
	ReAct	Near	**0.251**	**0.092**	0.094	**0.253**	**0.102**	0.097	**0.279**	**0.110**	0.111
		Far	**0.103**	**0.092**	0.094	**0.116**	**0.104**	0.098	**0.128**	**0.113**	0.112
	KNN	Near	**0.309**	0.116	0.115	**0.319**	0.125	0.118	**0.325**	0.133	0.127
		Far	**0.158**	**0.115**	0.115	**0.161**	**0.125**	0.119	**0.167**	**0.132**	0.127
	Ash	Near	**0.262**	**0.110**	0.108	**0.287**	**0.121**	0.122	**0.299**	**0.132**	0.135
		Far	**0.110**	**0.110**	0.108	**0.131**	**0.121**	0.122	**0.140**	**0.133**	0.136

Table 5.6: Estimation Error $(w - \hat{w})^2 / K(\downarrow)$ of our OSLS-MAP estimation model on ImageNet-200 dataset with Near OOD datasets and Far OOD datasets comparison under Ordered-LT (Forward) ID and OOD label shift. Settings in which our model outperforms baselines are colored in gray. Our model outperforms baselines under most label shift settings. Each metric is averaged among corresponding OOD test set (Tab. 5.1) and over three independent ID classifiers.

Setting			CIFAR10-LT10 Train, Long-Tailed Shifted Test, MAP								
ID label Shift param			LT10 Forward			LT50 Forward			LT100 Forward		
OOD label shift param r			1.0	0.1	0.01	1.0	0.1	0.01	1.0	0.1	0.01
Closed Set Label Shift estimation models											
BBSE		Near	0.087	0.020	0.013	0.287	0.248	0.252	0.419	0.358	0.450
		Far	0.371	0.030	0.013	0.633	0.260	0.252	0.742	0.370	0.453
MLLS		Near	0.135	0.021	0.013	0.268	0.015	0.006	0.341	0.013	0.004
		Far	0.628	0.033	0.013	0.807	0.023	0.006	0.795	0.018	0.003
RLLS		Near	0.000	0.000	0.000	0.207	0.206	0.206	0.328	0.327	0.327
		Far	0.000	0.000	0.000	0.206	0.206	0.206	0.328	0.327	0.327
MAPLS		Near	0.354	0.111	0.059	0.561	0.142	0.106	0.668	0.142	0.102
		Far	0.553	0.108	0.058	0.836	0.139	0.105	0.888	0.133	0.099
Open Set Label Shift estimation models											
Baseline			1.806	1.806	1.806	3.116	3.116	3.116	3.477	3.477	3.477
ours	OpenMax	Near	0.094	0.067	0.058	0.285	0.167	0.156	0.347	0.183	0.172
		Far	0.167	0.065	0.058	0.326	0.162	0.154	0.382	0.180	0.172
	MLS	Near	0.091	0.074	0.068	0.238	0.148	0.148	**0.265**	0.181	0.172
		Far	0.104	0.073	0.068	0.228	0.144	0.148	**0.250**	0.176	0.173
	ReAct	Near	0.037	0.027	0.037	**0.153**	0.127	0.121	**0.191**	0.162	0.156
		Far	0.051	0.027	0.037	**0.148**	0.125	0.120	**0.184**	0.159	0.156
	KNN	Near	0.152	0.091	0.095	0.307	0.170	0.163	0.356	0.202	0.198
		Far	0.178	0.089	0.094	0.310	0.166	0.162	0.356	0.199	0.197
	Ash	Near	0.044	0.027	0.026	**0.176**	0.129	0.121	**0.225**	0.173	0.152
		Far	0.084	0.027	0.026	**0.203**	0.129	0.121	**0.250**	0.168	0.152

Table 5.7: Estimation Error $(w - \hat{w})^2 / K(\downarrow)$ of our OSLS-MAP estimation model on the Long-Tailed CIFAR10 dataset with Near OOD datasets and Far OOD datasets comparison under different ID and OOD label shift. Outperforming results are in bold face and settings that outperform the baseline are colored in gray. Our model outperforms baseline under most label shift settings. Each metric is averaged among corresponding OOD test set (Tab. 5.1) and over three independent ID classifiers.

Setting			CIFAR10-LT10 Train, Long-Tailed Shifted Test, MAP								
ID label Shift param			LT10 Backward			LT50 Backward			LT100 Backward		
OOD label shift param r			1.0	0.1	0.01	1.0	0.1	0.01	1.0	0.1	0.01
Closed Set Label Shift estimation models											
BBSE		Near	11.416	10.884	10.371	21.239	18.996	18.659	25.916	22.358	23.182
		Far	11.606	10.843	10.374	21.299	18.977	18.659	25.983	22.375	23.170
MLLS		Near	2.494	0.375	0.569	5.374	0.893	0.779	6.029	1.477	0.841
		Far	3.663	0.421	0.575	6.633	0.974	0.787	7.523	1.558	0.851
RLLS		Near	11.638	11.614	11.615	23.617	23.568	23.565	29.429	29.383	29.351
		Far	11.639	11.614	11.615	23.620	23.569	23.566	29.432	29.383	29.351
MAPLS		Near	3.574	1.340	1.465	8.065	2.403	1.999	9.387	3.242	1.428
		Far	3.839	1.373	1.472	8.167	2.489	2.012	9.570	3.307	1.433
Open Set Label Shift estimation models											
Baseline			4.522	4.522	4.522	13.156	13.156	13.156	17.770	17.770	17.770
ours	OpenMax	Near	2.651	0.875	**0.676**	**5.855**	1.984	1.550	**7.314**	2.349	1.865
		Far	**2.823**	0.880	0.673	**5.925**	1.987	1.548	**7.502**	2.412	1.863
	MLS	Near	**1.156**	**0.489**	**0.416**	**2.575**	**1.087**	**1.037**	**3.538**	1.617	**1.376**
		Far	**1.001**	**0.469**	**0.415**	**2.259**	**1.059**	**1.042**	**3.025**	1.566	**1.376**
	ReAct	Near	4.070	3.317	3.281	8.738	7.485	7.300	11.338	9.603	9.291
		Far	3.990	3.314	3.281	8.478	7.436	7.290	10.998	9.586	9.279
	KNN	Near	**1.243**	**0.432**	**0.350**	**2.967**	**1.088**	0.911	**4.016**	**1.370**	**1.283**
		Far	**1.057**	**0.417**	**0.344**	**2.547**	**1.026**	0.906	**3.312**	**1.330**	**1.272**
	Ash	Near	6.301	5.001	4.862	12.779	10.446	10.248	15.997	13.032	12.664
		Far	6.196	4.983	4.862	12.429	10.400	10.255	15.463	12.977	12.654

Table 5.8: Estimation Error $(w - \hat{w})^2 / K(\downarrow)$ of our OSLS-MAP estimation model on the Long-Tailed CIFAR10 dataset with Near OOD datasets and Far OOD datasets comparison under different ID and OOD label shift. Outperforming results are in bold face and settings that outperform the baseline are colored in gray. Our model outperforms baseline under most label shift settings. Each metric is averaged among corresponding OOD test set (Tab. 5.1) and over three independent ID classifiers.

Setting			CIFAR100-LT10 Train, Long-Tailed Shifted Test, MAP								
ID label Shift param			LT10 Forward			LT50 Forward			LT100 Forward		
OOD label shift param r			1.0	0.1	0.01	1.0	0.1	0.01	1.0	0.1	0.01
Closed Set Label Shift estimation models											
BBSE		Near	1.184	0.996	1.109	1.926	1.409	1.615	1.693	1.974	2.646
		Far	4.498	1.230	1.081	4.306	1.931	1.531	3.861	2.117	2.518
MLLS		Near	143.984	24.795	36.063	207.347	29.882	14.939	203.636	18.749	6.636
		Far	367.633	42.241	37.423	225.657	45.362	14.434	700.636	23.536	6.809
RLLS		Near	0.001	0.001	0.001	0.209	0.206	0.205	0.316	0.313	0.313
		Far	0.002	0.001	0.001	0.212	0.207	0.206	0.323	0.313	0.313
MAPLS		Near	50.587	11.288	15.579	86.141	14.844	10.252	76.854	9.392	5.319
		Far	193.608	16.395	15.934	121.326	20.920	10.035	419.148	11.608	5.350
Open Set Label Shift estimation models											
Baseline			1.317	1.317	1.317	2.445	2.445	2.445	2.739	2.739	2.739
ours	OpenMax	Near	0.223	0.049	0.046	0.234	**0.069**	**0.063**	**0.254**	**0.082**	**0.083**
		Far	0.474	0.055	0.047	0.490	**0.073**	**0.063**	0.475	**0.085**	**0.083**
	MLS	Near	0.360	0.098	0.094	0.337	**0.127**	**0.113**	0.408	**0.127**	**0.114**
		Far	1.385	0.115	0.095	1.028	**0.146**	**0.114**	1.219	**0.144**	**0.115**
	ReAct	Near	0.337	0.080	0.080	0.372	**0.103**	**0.087**	0.374	**0.106**	**0.091**
		Far	1.132	0.093	0.080	1.045	**0.111**	**0.087**	0.943	**0.110**	**0.092**
	KNN	Near	0.630	0.139	0.137	0.622	**0.137**	**0.123**	0.560	**0.137**	**0.124**
		Far	0.659	0.150	0.137	0.665	**0.141**	**0.122**	0.641	**0.143**	**0.124**
	Ash	Near	0.380	0.081	0.086	0.423	**0.103**	**0.096**	0.414	**0.108**	**0.100**
		Far	1.197	0.095	0.086	1.109	**0.115**	**0.096**	1.099	**0.116**	**0.100**

Table 5.9: Estimation Error $(w - \hat{w})^2 / K(\downarrow)$ of our OSLS-MAP estimation model on the Long-Tailed CIFAR100 dataset with Near OOD datasets and Far OOD datasets comparison under different ID and OOD label shift. Outperforming results are in bold face and settings that outperform the baseline are colored in gray. Our model outperforms baseline under most label shift settings. Each metric is averaged among corresponding OOD test set (Tab. 5.1) and over three independent ID classifiers.

Setting		CIFAR100-LT10 Train, Long-Tailed Shifted Test, MAP								
ID label Shift param		LT10 Backward			LT50 Backward			LT100 Backward		
OOD label shift param r		1.0	0.1	0.01	1.0	0.1	0.01	1.0	0.1	0.01
Closed Set Label Shift estimation models										
BBSE	Near	9.860	9.465	9.747	18.756	18.156	18.321	22.142	21.652	22.186
	Far	12.680	9.544	9.744	21.578	18.121	18.336	23.909	21.613	22.183
MLLS	Near	222.536	88.341	60.125	218.213	152.682	96.968	466.629	104.969	171.657
	Far	1229.442	97.308	59.798	496.811	154.811	97.943	950.108	99.563	176.237
RLLS	Near	8.091	8.058	8.025	16.896	16.883	16.755	21.007	20.940	20.831
	Far	8.087	8.063	8.025	16.891	16.878	16.754	20.951	20.946	20.830
MAPLS	Near	87.014	32.047	23.384	100.514	67.379	47.580	229.440	48.819	82.531
	Far	740.536	36.744	23.326	297.858	71.695	48.028	564.773	47.701	84.695
Open Set Label Shift estimation models										
Baseline		3.090	3.090	3.090	9.471	9.471	9.471	12.774	12.774	12.774
ours OpenMax	Near	5.285	3.895	3.738	10.646	**7.620**	**7.362**	12.826	**9.585**	**8.734**
	Far	5.429	3.837	3.729	10.144	**7.474**	**7.352**	**12.148**	**9.335**	**8.711**
ours MLS	Near	4.166	3.296	3.180	**8.431**	**6.692**	**6.642**	**10.218**	**7.896**	**8.325**
	Far	5.397	3.337	3.182	9.560	**6.704**	**6.634**	**11.348**	**7.936**	**8.335**
ours ReAct	Near	5.050	3.865	3.781	9.637	**7.875**	**7.747**	**12.020**	**9.642**	**9.581**
	Far	6.012	3.883	3.777	10.705	**7.915**	**7.749**	**12.763**	**9.617**	**9.578**
ours KNN	Near	6.005	4.948	5.101	11.816	10.814	10.702	14.690	12.925	13.255
	Far	6.175	4.968	5.106	11.987	10.867	10.711	14.784	12.989	13.268
ours Ash	Near	4.775	3.548	3.408	9.493	**7.084**	**6.825**	**11.573**	**8.735**	**8.344**
	Far	5.906	3.527	3.409	10.197	**6.999**	**6.819**	**12.438**	**8.681**	**8.326**

Table 5.10: Estimation Error $(w - \hat{w})^2/K(\downarrow)$ of our OSLS-MAP estimation model on the Long-Tailed CIFAR100 dataset with Near OOD datasets and Far OOD datasets comparison under different ID and OOD label shift. Outperforming results are in bold face and settings that outperform the baseline are colored in gray. Our model outperforms baseline under most label shift settings. Each metric is averaged among corresponding OOD test set (Tab. 5.1) and over three independent ID classifiers.

5.4.3 Ablation Study

Assumption Analysis: Assumption 5A used in the model is a common assumption for Neural Network classifiers, which has been used in previous label shift estimation problem (Ye et al., 2024) and other classification tasks like calibration (Liang et al., 2017) and Long-Tailed Recognition (Xu et al., 2021). We justify Assumption 5A with empirical evidence from the practical classifiers used in our experiments.

As discussed in the previous works (Garg, Wu, Balakrishnan et al., 2020), if Assumption 5A is satisfied, classifier f is a perfectly calibrated classifier on the source domain. The calibration performance of the classifier is commonly evaluated via the Expected Calibration Error (ECE) (Guo, Pleiss et al., 2017; Liu, Ye, Cui et al., 2024; Liu, Ye, Wang et al., 2023), where a well-calibrated classifier will have ECE close to 0. In this work, we provide the calibration performance of the practical classifiers f used in our model.

As shown in Tab. 5.11, the classifiers that we used in our model have good calibration performance. Hence Assumption 5A is likely to be satisfied, which justifies the practical applicability of our model in the real world problems.

Performance Impact Factors: Our proposed model depends on several assumptions (Assumption 1, Assumption 5) and the availability of the source and target domain datasets $\mathcal{D}^s, \mathcal{D}^t$. In practice, even if these assumptions are satisfied, several other factors could influence the performance of our OSLS-MLE/MAP model.

Dataset	Classifier 1	Classifier 2	Classifier 3
CIFAR10	0.0277	0.0281	0.0242
CIFAR100	0.0628	0.0600	0.0621
ImageNet-200	0.0163	0.0133	0.0131

Table 5.11: Calibration Performance in terms of ECE (↓) of the ID classifiers (3 for each dataset) used in our model on the source domain validation set. The classifiers has good calibration performance and therefore Assumption 5 is likely to be satisfied in our practical model.

Theoretically speaking, our model requires i.i.d. samples in the datasets $\mathcal{D}^s, \mathcal{D}^t, \mathcal{D}^o$ and thus violation of this requirement will introduce extra estimation error. Secondly, the noise in the ground truth labels of the source domain train dataset could affect the estimate of the source label distribution, and thus introduce extra estimation error. On the other hand, it is also worth noting that, using data augmentation techniques (*e.g.* mixup, noise label) when training the ID classifier f does not influence the validity of the proposed model.

Empirically speaking, we observe that several other factors could impact the performance of the proposed OSLS-MLE/MAP model and other Closed Set Label Shift models:

- **Amount of label shift**: As shown in Tab. 5.10, the estimation error of the label shift models (including ours) can degrade significantly when the source and target domain have highly imbalanced label distribution.
- **ID/OOD ratio**: As shown in Tab. 5.5, the estimation error of our model also decreases when more OOD data presents in the test set.
- **OOD classifier**: As shown in Tab. 5.3, the Top1 Accuracy of our model with different OOD classifier could vary significantly.

Pseudo OOD data Generation: We also provide the sensitivity analysis of the hyperparameter γ in Eq. (5.18) using in our OSLS-MLE model when generating pseudo OOD samples with Gaussian noise:

$$\mathcal{D}^{\mathbf{o}}_{\gamma} = \{(1-\gamma)\cdot x_i + \gamma \cdot \epsilon | x_i \in \mathcal{D}^s, \epsilon \sim \mathcal{N}(0,1)\}. \tag{5.21}$$

As shown in Tab. 5.12, 5.13, our model exhibits stable performance when γ varies.

Training Time analysis: The training time of our OSLS *estimation* model with a NVIDIA RTX 2080Ti GPU on CIFAR10/100 dataset is less than 1 second, on ImageNet-200 dataset is less than 5 seconds and on ImageNet 1k is about 20 seconds. Since EM algorithm is scalable to large scale datasets Ye et al. (2024), our model can be easily applied to real world problems with affordable computational overhead.

Dataset			CIFAR100								
ID label Shift param			LT10 Forward			LT50 Forward			LT100 Forward		
OOD label shift param r			1.0	0.1	0.01	1.0	0.1	0.01	1.0	0.1	0.01
Closed Set Label Shift estimation models											
BBSE		Near	$0.560_{\pm 0.038}$	$0.121_{\pm 0.027}$	$0.107_{\pm 0.026}$	$0.758_{\pm 0.057}$	$0.171_{\pm 0.044}$	$0.136_{\pm 0.031}$	$0.841_{\pm 0.038}$	$0.188_{\pm 0.048}$	$0.151_{\pm 0.030}$
		Far	$4.128_{\pm 0.245}$	$0.253_{\pm 0.028}$	$0.109_{\pm 0.027}$	$4.370_{\pm 0.301}$	$0.291_{\pm 0.043}$	$0.139_{\pm 0.034}$	$4.431_{\pm 0.228}$	$0.306_{\pm 0.054}$	$0.153_{\pm 0.031}$
MLLS		Near	$0.906_{\pm 0.061}$	$0.114_{\pm 0.028}$	$0.088_{\pm 0.028}$	$1.029_{\pm 0.066}$	$0.155_{\pm 0.044}$	$0.105_{\pm 0.028}$	$1.072_{\pm 0.067}$	$0.150_{\pm 0.042}$	$0.113_{\pm 0.034}$
		Far	$9.633_{\pm 1.442}$	$0.348_{\pm 0.057}$	$0.092_{\pm 0.028}$	$9.910_{\pm 1.551}$	$0.380_{\pm 0.044}$	$0.108_{\pm 0.030}$	$9.896_{\pm 1.523}$	$0.373_{\pm 0.065}$	$0.115_{\pm 0.035}$
RLLS		Near	$0.426_{\pm 0.000}$	$0.425_{\pm 0.000}$	$0.425_{\pm 0.000}$	$1.100_{\pm 0.000}$	$1.099_{\pm 0.000}$	$1.099_{\pm 0.000}$	$1.404_{\pm 0.000}$	$1.402_{\pm 0.000}$	$1.402_{\pm 0.000}$
		Far	$0.426_{\pm 0.000}$	$0.425_{\pm 0.000}$	$0.425_{\pm 0.000}$	$1.100_{\pm 0.000}$	$1.099_{\pm 0.000}$	$1.099_{\pm 0.000}$	$1.404_{\pm 0.000}$	$1.403_{\pm 0.000}$	$1.402_{\pm 0.000}$
MAPLS		Near	$0.700_{\pm 0.034}$	$0.114_{\pm 0.018}$	$0.091_{\pm 0.019}$	$0.878_{\pm 0.037}$	$0.164_{\pm 0.033}$	$0.120_{\pm 0.019}$	$0.946_{\pm 0.041}$	$0.175_{\pm 0.031}$	$0.135_{\pm 0.026}$
		Far	$7.469_{\pm 1.122}$	$0.290_{\pm 0.040}$	$0.094_{\pm 0.018}$	$7.758_{\pm 1.196}$	$0.340_{\pm 0.029}$	$0.123_{\pm 0.020}$	$7.779_{\pm 1.171}$	$0.350_{\pm 0.046}$	$0.138_{\pm 0.026}$
Open Set Label Shift estimation models											
Baseline			$0.426_{\pm 0.000}$	$0.426_{\pm 0.000}$	$0.426_{\pm 0.000}$	$1.101_{\pm 0.000}$	$1.101_{\pm 0.000}$	$1.101_{\pm 0.000}$	$1.405_{\pm 0.000}$	$1.405_{\pm 0.000}$	$1.405_{\pm 0.000}$
ours	$\gamma = 0.1$	Near	$\mathbf{0.396}_{\pm 0.018}$	$\mathbf{0.043}_{\pm 0.004}$	$\mathbf{0.046}_{\pm 0.009}$	$\mathbf{0.470}_{\pm 0.027}$	$\mathbf{0.068}_{\pm 0.013}$	$\mathbf{0.073}_{\pm 0.013}$	$\mathbf{0.508}_{\pm 0.005}$	$\mathbf{0.085}_{\pm 0.017}$	$\mathbf{0.078}_{\pm 0.015}$
		Far	$2.152_{\pm 0.396}$	$\mathbf{0.082}_{\pm 0.008}$	$\mathbf{0.047}_{\pm 0.009}$	$2.224_{\pm 0.323}$	$\mathbf{0.104}_{\pm 0.010}$	$\mathbf{0.074}_{\pm 0.013}$	$2.426_{\pm 0.329}$	$\mathbf{0.118}_{\pm 0.012}$	$\mathbf{0.079}_{\pm 0.015}$
	$\gamma = 0.2$	Near	$0.473_{\pm 0.008}$	$\mathbf{0.039}_{\pm 0.005}$	$\mathbf{0.032}_{\pm 0.001}$	$\mathbf{0.573}_{\pm 0.023}$	$\mathbf{0.062}_{\pm 0.002}$	$\mathbf{0.051}_{\pm 0.004}$	$\mathbf{0.589}_{\pm 0.013}$	$\mathbf{0.063}_{\pm 0.008}$	$\mathbf{0.057}_{\pm 0.005}$
		Far	$3.100_{\pm 0.115}$	$\mathbf{0.094}_{\pm 0.007}$	$\mathbf{0.034}_{\pm 0.001}$	$3.077_{\pm 0.221}$	$\mathbf{0.111}_{\pm 0.011}$	$\mathbf{0.053}_{\pm 0.004}$	$3.144_{\pm 0.221}$	$\mathbf{0.118}_{\pm 0.009}$	$\mathbf{0.058}_{\pm 0.005}$
	$\gamma = 0.3$	Near	$0.480_{\pm 0.031}$	$\mathbf{0.034}_{\pm 0.002}$	$\mathbf{0.034}_{\pm 0.002}$	$\mathbf{0.543}_{\pm 0.022}$	$\mathbf{0.062}_{\pm 0.008}$	$\mathbf{0.055}_{\pm 0.008}$	$\mathbf{0.601}_{\pm 0.016}$	$\mathbf{0.057}_{\pm 0.008}$	$\mathbf{0.056}_{\pm 0.011}$
		Far	$3.069_{\pm 0.082}$	$\mathbf{0.089}_{\pm 0.004}$	$\mathbf{0.036}_{\pm 0.002}$	$3.268_{\pm 0.238}$	$\mathbf{0.116}_{\pm 0.010}$	$\mathbf{0.057}_{\pm 0.008}$	$3.236_{\pm 0.104}$	$\mathbf{0.114}_{\pm 0.008}$	$\mathbf{0.057}_{\pm 0.012}$
	$\gamma = 0.4$	Near	$0.482_{\pm 0.039}$	$\mathbf{0.035}_{\pm 0.002}$	$\mathbf{0.032}_{\pm 0.004}$	$\mathbf{0.571}_{\pm 0.027}$	$\mathbf{0.067}_{\pm 0.012}$	$\mathbf{0.056}_{\pm 0.010}$	$\mathbf{0.609}_{\pm 0.027}$	$\mathbf{0.062}_{\pm 0.009}$	$\mathbf{0.053}_{\pm 0.003}$
		Far	$3.093_{\pm 0.068}$	$\mathbf{0.100}_{\pm 0.007}$	$\mathbf{0.034}_{\pm 0.004}$	$3.272_{\pm 0.117}$	$\mathbf{0.123}_{\pm 0.010}$	$\mathbf{0.057}_{\pm 0.010}$	$3.314_{\pm 0.076}$	$\mathbf{0.123}_{\pm 0.002}$	$\mathbf{0.054}_{\pm 0.002}$
	$\gamma = 0.5$	Near	$0.486_{\pm 0.025}$	$\mathbf{0.041}_{\pm 0.003}$	$\mathbf{0.032}_{\pm 0.004}$	$\mathbf{0.578}_{\pm 0.017}$	$\mathbf{0.063}_{\pm 0.013}$	$\mathbf{0.055}_{\pm 0.007}$	$\mathbf{0.598}_{\pm 0.028}$	$\mathbf{0.060}_{\pm 0.007}$	$\mathbf{0.055}_{\pm 0.008}$
		Far	$3.135_{\pm 0.155}$	$\mathbf{0.102}_{\pm 0.007}$	$\mathbf{0.033}_{\pm 0.003}$	$3.209_{\pm 0.135}$	$\mathbf{0.131}_{\pm 0.010}$	$\mathbf{0.057}_{\pm 0.008}$	$3.335_{\pm 0.123}$	$\mathbf{0.115}_{\pm 0.008}$	$\mathbf{0.056}_{\pm 0.008}$

Table 5.12: Ablation study of hyperparameter γ when generating pseudo OOD samples. Estimation Error $(w - \hat{w})^2/K(\downarrow)$ of our OSLS estimation model (OpenMax OOD detector) on CIFAR100 dataset with Near OOD datasets and Far OOD datasets comparison under Ordered-LT (Forward) ID and OOD label shift.

Dataset			CIFAR100								
ID label Shift param			LT10 Backward			LT50 Backward			LT100 Backward		
OOD label shift param r			1.0	0.1	0.01	1.0	0.1	0.01	1.0	0.1	0.01
Closed Set Label Shift estimation models											
BBSE		Near	$0.540_{\pm 0.029}$	$0.152_{\pm 0.008}$	$0.159_{\pm 0.025}$	$0.732_{\pm 0.050}$	$0.252_{\pm 0.027}$	$0.264_{\pm 0.035}$	$0.778_{\pm 0.026}$	$0.281_{\pm 0.050}$	$0.339_{\pm 0.073}$
		Far	$4.042_{\pm 0.273}$	$0.276_{\pm 0.011}$	$0.161_{\pm 0.023}$	$4.075_{\pm 0.388}$	$0.381_{\pm 0.049}$	$0.262_{\pm 0.037}$	$4.080_{\pm 0.223}$	$0.387_{\pm 0.056}$	$0.339_{\pm 0.077}$
MLLS		Near	$0.912_{\pm 0.083}$	$0.131_{\pm 0.013}$	$0.119_{\pm 0.009}$	$1.107_{\pm 0.085}$	$0.203_{\pm 0.012}$	$0.173_{\pm 0.017}$	$1.152_{\pm 0.061}$	$0.218_{\pm 0.017}$	$0.203_{\pm 0.022}$
		Far	$9.500_{\pm 1.553}$	$0.332_{\pm 0.036}$	$0.118_{\pm 0.008}$	$9.583_{\pm 1.578}$	$0.404_{\pm 0.058}$	$0.167_{\pm 0.017}$	$9.494_{\pm 1.499}$	$0.381_{\pm 0.039}$	$0.201_{\pm 0.024}$
RLLS		Near	$0.426_{\pm 0.000}$	$0.425_{\pm 0.000}$	$0.425_{\pm 0.000}$	$1.100_{\pm 0.000}$	$1.099_{\pm 0.000}$	$1.099_{\pm 0.000}$	$1.403_{\pm 0.000}$	$1.402_{\pm 0.000}$	$1.402_{\pm 0.000}$
		Far	$0.426_{\pm 0.000}$	$0.425_{\pm 0.000}$	$0.425_{\pm 0.000}$	$1.100_{\pm 0.000}$	$1.099_{\pm 0.000}$	$1.099_{\pm 0.000}$	$1.403_{\pm 0.000}$	$1.402_{\pm 0.000}$	$1.402_{\pm 0.000}$
MAPLS		Near	$0.710_{\pm 0.052}$	$0.119_{\pm 0.007}$	$0.106_{\pm 0.003}$	$0.941_{\pm 0.063}$	$0.196_{\pm 0.008}$	$0.159_{\pm 0.009}$	$1.007_{\pm 0.044}$	$0.218_{\pm 0.012}$	$0.188_{\pm 0.012}$
		Far	$7.360_{\pm 1.206}$	$0.268_{\pm 0.025}$	$0.106_{\pm 0.002}$	$7.476_{\pm 1.220}$	$0.345_{\pm 0.039}$	$0.155_{\pm 0.009}$	$7.439_{\pm 1.153}$	$0.339_{\pm 0.023}$	$0.186_{\pm 0.012}$
Open Set Label Shift estimation models											
Baseline			$0.426_{\pm 0.000}$	$0.426_{\pm 0.000}$	$0.426_{\pm 0.000}$	$1.101_{\pm 0.000}$	$1.101_{\pm 0.000}$	$1.101_{\pm 0.000}$	$1.405_{\pm 0.000}$	$1.405_{\pm 0.000}$	$1.405_{\pm 0.000}$
ours	$\gamma = 0.1$	Near	$0.428_{\pm 0.046}$	$\mathbf{0.041}_{\pm 0.002}$	$\mathbf{0.034}_{\pm 0.004}$	$\mathbf{0.565}_{\pm 0.076}$	$\mathbf{0.058}_{\pm 0.001}$	$\mathbf{0.056}_{\pm 0.006}$	$\mathbf{0.565}_{\pm 0.028}$	$\mathbf{0.063}_{\pm 0.001}$	$\mathbf{0.052}_{\pm 0.003}$
		Far	$2.105_{\pm 0.489}$	$\mathbf{0.080}_{\pm 0.007}$	$\mathbf{0.035}_{\pm 0.003}$	$2.244_{\pm 0.372}$	$\mathbf{0.093}_{\pm 0.004}$	$\mathbf{0.058}_{\pm 0.007}$	$2.192_{\pm 0.320}$	$\mathbf{0.100}_{\pm 0.015}$	$\mathbf{0.054}_{\pm 0.003}$
	$\gamma = 0.2$	Near	$0.497_{\pm 0.025}$	$\mathbf{0.034}_{\pm 0.004}$	$\mathbf{0.028}_{\pm 0.001}$	$\mathbf{0.628}_{\pm 0.017}$	$\mathbf{0.059}_{\pm 0.008}$	$\mathbf{0.049}_{\pm 0.005}$	$\mathbf{0.664}_{\pm 0.025}$	$\mathbf{0.058}_{\pm 0.007}$	$\mathbf{0.046}_{\pm 0.005}$
		Far	$2.879_{\pm 0.178}$	$\mathbf{0.093}_{\pm 0.007}$	$\mathbf{0.029}_{\pm 0.001}$	$2.855_{\pm 0.192}$	$\mathbf{0.115}_{\pm 0.019}$	$\mathbf{0.050}_{\pm 0.004}$	$2.879_{\pm 0.215}$	$\mathbf{0.115}_{\pm 0.010}$	$\mathbf{0.047}_{\pm 0.005}$
	$\gamma = 0.3$	Near	$0.539_{\pm 0.012}$	$\mathbf{0.034}_{\pm 0.004}$	$\mathbf{0.029}_{\pm 0.006}$	$\mathbf{0.662}_{\pm 0.014}$	$\mathbf{0.060}_{\pm 0.010}$	$\mathbf{0.048}_{\pm 0.003}$	$\mathbf{0.688}_{\pm 0.021}$	$\mathbf{0.066}_{\pm 0.007}$	$\mathbf{0.053}_{\pm 0.003}$
		Far	$3.000_{\pm 0.185}$	$\mathbf{0.092}_{\pm 0.015}$	$\mathbf{0.030}_{\pm 0.006}$	$2.987_{\pm 0.171}$	$\mathbf{0.113}_{\pm 0.006}$	$\mathbf{0.050}_{\pm 0.003}$	$3.025_{\pm 0.235}$	$\mathbf{0.114}_{\pm 0.022}$	$\mathbf{0.055}_{\pm 0.003}$
	$\gamma = 0.4$	Near	$0.534_{\pm 0.001}$	$\mathbf{0.037}_{\pm 0.003}$	$\mathbf{0.027}_{\pm 0.002}$	$\mathbf{0.669}_{\pm 0.024}$	$\mathbf{0.059}_{\pm 0.002}$	$\mathbf{0.049}_{\pm 0.001}$	$\mathbf{0.696}_{\pm 0.019}$	$\mathbf{0.063}_{\pm 0.009}$	$\mathbf{0.046}_{\pm 0.002}$
		Far	$3.001_{\pm 0.144}$	$\mathbf{0.101}_{\pm 0.009}$	$\mathbf{0.028}_{\pm 0.003}$	$3.117_{\pm 0.085}$	$\mathbf{0.113}_{\pm 0.014}$	$\mathbf{0.050}_{\pm 0.001}$	$3.023_{\pm 0.134}$	$\mathbf{0.131}_{\pm 0.016}$	$\mathbf{0.048}_{\pm 0.003}$
	$\gamma = 0.5$	Near	$0.526_{\pm 0.018}$	$\mathbf{0.035}_{\pm 0.002}$	$\mathbf{0.030}_{\pm 0.004}$	$\mathbf{0.647}_{\pm 0.005}$	$\mathbf{0.052}_{\pm 0.004}$	$\mathbf{0.054}_{\pm 0.005}$	$\mathbf{0.693}_{\pm 0.017}$	$\mathbf{0.060}_{\pm 0.002}$	$\mathbf{0.044}_{\pm 0.004}$
		Far	$3.078_{\pm 0.159}$	$\mathbf{0.092}_{\pm 0.017}$	$\mathbf{0.032}_{\pm 0.003}$	$3.046_{\pm 0.153}$	$\mathbf{0.122}_{\pm 0.018}$	$\mathbf{0.056}_{\pm 0.006}$	$3.021_{\pm 0.160}$	$\mathbf{0.118}_{\pm 0.021}$	$\mathbf{0.046}_{\pm 0.005}$

Table 5.13: Ablation study of hyperparameter γ when generating pseudo OOD samples. Estimation Error $(w - \hat{w})^2/K(\downarrow)$ of our OSLS estimation model (OpenMax OOD detector) on the CIFAR100 dataset with Near OOD datasets and Far OOD datasets comparison under Ordered-LT (Backward) ID and OOD label shift.

5.5 Conclusion

In this chapter, we analyze the problem of Open Set Label Shift and propose a model to estimate the target label distribution of ID and OOD class. Then, we adapt a source domain ID/OOD classifier to the target domain with our retraining. With reasonable assumptions and an OOD reference dataset, our estimate of target label distribution is built on three estimates: 1) an estimate of the source label distribution of the ID/OOD class, 2) an estimate of the target label distribution of the ID/OOD class and 3) an estimate of the target label distribution of the OOD class when some assumptions on the OOD classifier are not satisfied. Based on the estimation results, the source domain ID/OOD classifier is then adapted to the target domain. Experiments on benchmark image classification datasets CIFAR10/100 and ImageNet-200 with different OOD datasets and label shift settings demonstrate the effectiveness of our model.

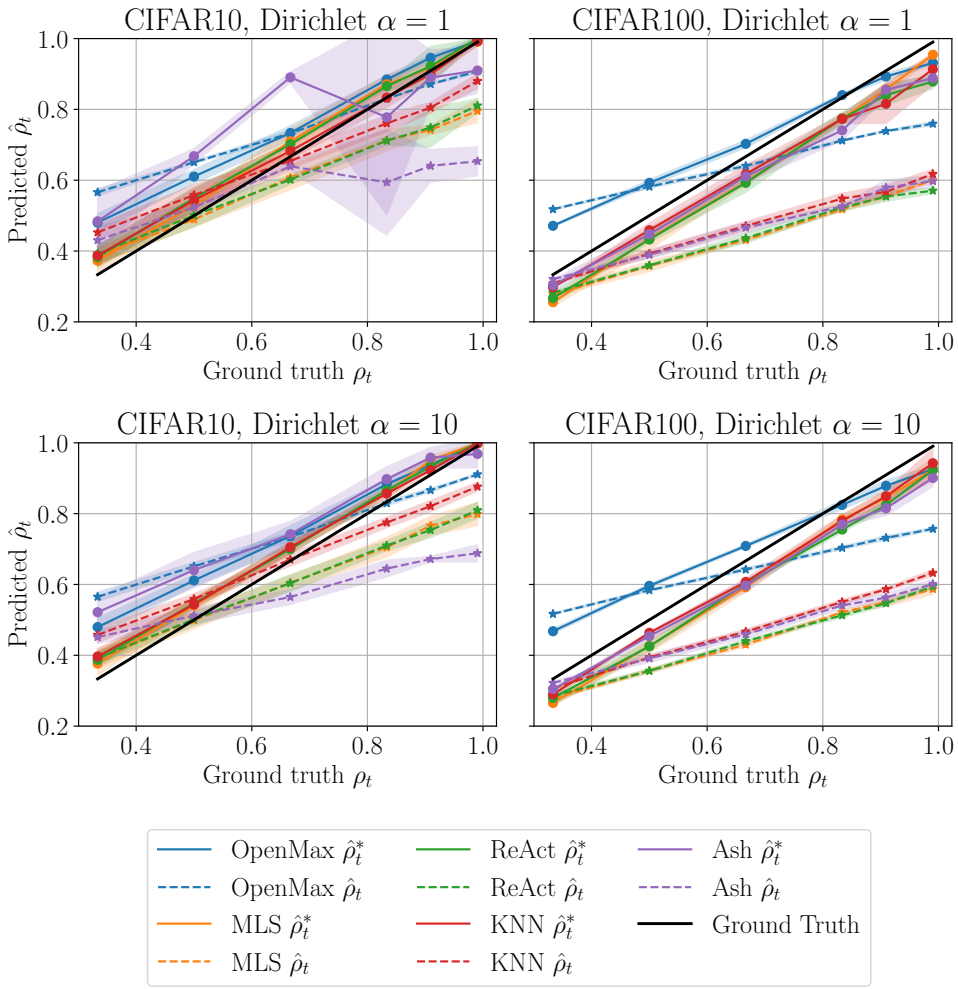

Figure 5.4: Estimation result comparison of $\hat{\rho}_t^*$ by our model (Solid lines), $\hat{\rho}_t$ by our model but without ρ_t correction (Section 5.3.4) (Dashed lines) based on different OOD classifiers and the Ground truth ρ_t (Black, Solid line), on CIFAR10/100 dataset with Dirichlet shift and Near OOD dataset (Tab. 5.1). The estimation result exhibits a linear correlation with the ground truth, which is explained by our analysis in Theorem 11 Moreover, our ρ_t correction model is able to adjust the predicted $\hat{\rho}_t$ to $\hat{\rho}_t^*$ that is closer to the ground truth. Shaded areas are $\pm$ one standard deviation over three independent ID classifiers.

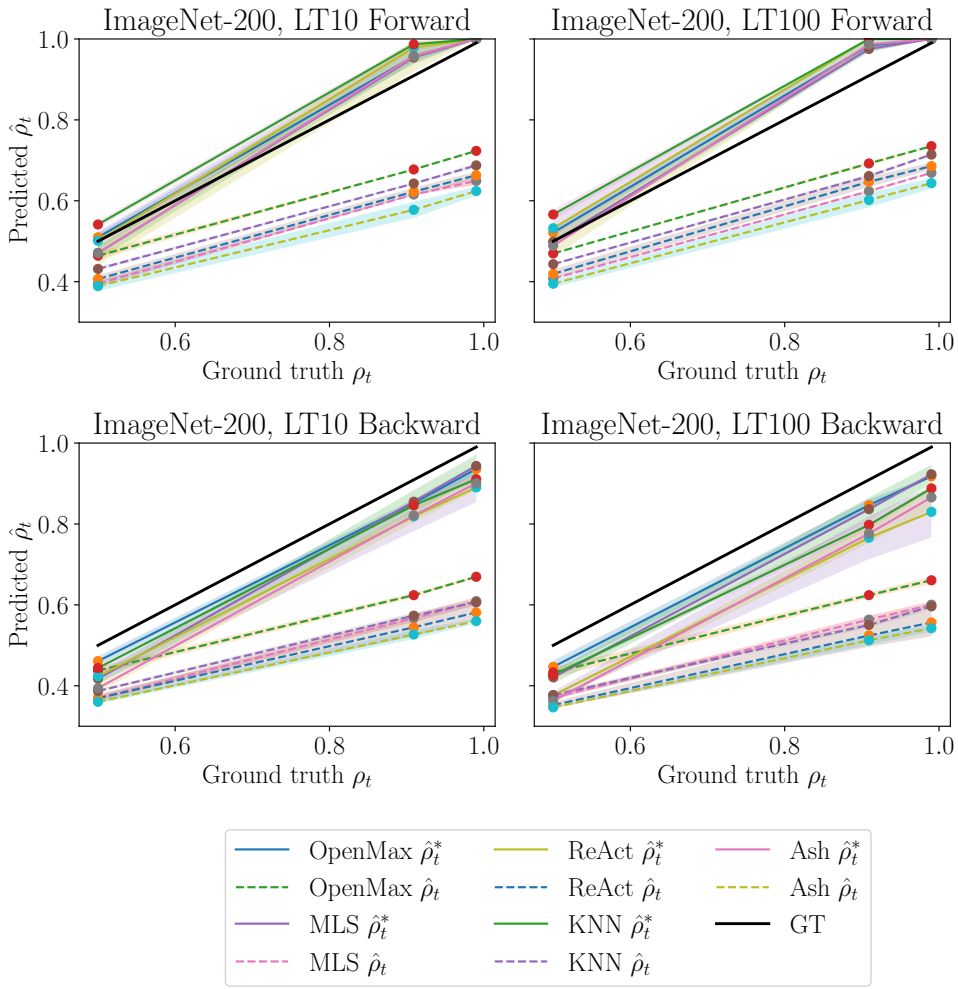

Figure 5.5: Estimation result comparison of $\hat{\rho}_t^*$ by our model (Solid lines), $\hat{\rho}_t$ by our model but without ρ_t correction (Section 5.3.4) (Dashed lines) based on different OOD classifiers and the Ground truth ρ_t (Black, Solid line), on ImageNet-200 dataset with LT10/LT100 shift and Near + Far OOD dataset (Tab. 5.1). Shaded areas are $\pm$ one standard deviation over corresponding OOD datasets and three independent ID classifiers.

Chapter 6

Label Shift Correction for Zero-Shot Learning

6.1 Introduction

This chapter considers the label shift problem in the Zero-Shot Learning (ZSL) task, which is more challenging compared with the Open Set Label Shift problem discussed in Chapter 5. Due to the challenging setting of Zero-Shot classification, we relax the label shift correction problem in the ZSL setting by assuming the target domain has uniform label distribution over both seen and unseen classes.

Refresher on the ZSL problem *Zero-Shot Learning* requires a model to be trained on images that show examples from one set of classes, referred to as *seen classes*, while being tested on images that show examples from another set of classes, referred to as *unseen classes*. During training, semantic information for both seen and unseen classes is provided to help infer the appearance of unseen classes.

ZSL literature in brief In the Zero-Shot Learning literature, many previous works, such as Changpinyo et al. (2016); Huang, Wang, Yu et al. (2019); Norouzi et al. (2013); Verma and Rai (2017); Xian, Lorenz et al. (2018), focus on learning a mapping between image features depicting certain classes and their corresponding semantic vectors. GFZSL (Verma and Rai, 2017) proposed a model similar to Kernel Ridge Regression to predict image features of unseen classes. GDAN (Huang, Wang, Yu et al., 2019) and f-CLSWGAN (Xian, Lorenz et al., 2018) utilize generative models like GAN (Goodfellow, Pouget-Abadie et al., 2014) and VAE (Kingma and Welling, 2013) to achieve the same objective. Based on these approaches, recent papers further learn a Neural Network (NN) projection from image feature space to a latent embedding space, where inter-class features can be better separated within each ZSL dataset (Chen, Wang et al., 2021; Han, Fu, Chen et al., 2021; Han, Fu and Yang, 2020; Le Cacheux et al., 2019b; Min et al., 2020). For example, in Han, Fu and Yang (2020), image features are projected to a latent space to “remove redundant information". FREE (Chen, Wang et al., 2021) adopts the same structure for “feature refinement" purposes. CE-GZSL (Han, Fu, Chen et al., 2021) also proposes a similar approach to generate a “contrastive embedding" of image features.

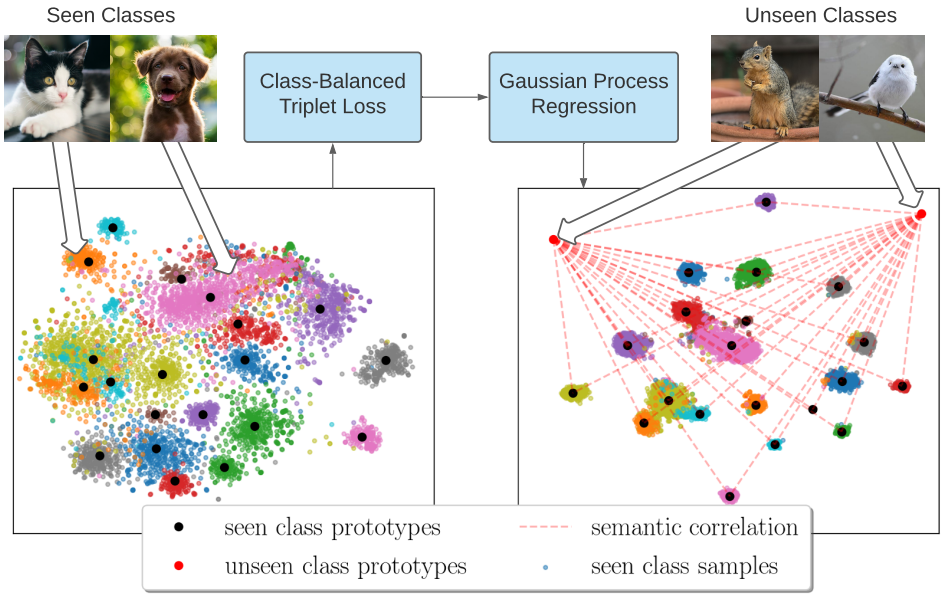

Figure 6.1: Overview our Class-Balanced Zero-Shot Learning model. We first train a latent embedding model for image features. The model is trained with Class-Balanced Triplet loss to separate inter-class features, which is robust to class-imbalanced datasets. Then, a Gaussian Process Regression model is proposed to predict unseen class prototypes based on seen class prototypes and semantic correlations between classes. Finally, our ZSL classifier is constructed based on the prototypes.

Motivations Firstly, the label shift problem or the class imbalance problem is rarely studied in the Zero-Shot Learning literature, even when the existing ZSL datasets have highly imbalanced source label distribution. For instance, the APY (Farhadi et al., 2009) training dataset has nearly $1/3$ of samples from the same class. AWA2 (Xian, Lampert et al., 2019) has 1645 samples in one class and only 100 samples in another in the training set. The label shift problem is not negligible when training a classification model on these datasets. Moreover, real world visual datasets usually exhibit a highly imbalanced or Long-Tailed label distribution (Liu, Miao, Zhan, Wang, Gong and Yu, 2019), which can have a significant impact on the performance of the classification models trained on these datasets (Buda et al., 2018a; Cui et al., 2019). Therefore, it is important to ensure that the ZSL models can handle the label shift problem when deploying these models to real world problems.

Contributions In this chapter, we propose a ZSL model that corrects label shift by assuming the target domain has a uniform label distribution over both seen and unseen classes. We propose a class-balanced triplet loss that separates image features in a latent embedding space for a class-imbalanced dataset. We also propose a Gaussian Process (GP) model to learn a mapping between the feature and semantic space. When used in the regression setting, the classical Gaussian Process (GP) is robust to overfitting (Rasmussen et al., 2006). If training and testing data come from the same distribution, a PAC-Bayesian Bound (Suzuki, 2012) guarantees that the training error will be close to the testing error. Our experiments demonstrate that our model, though employing a simple design, can achieve state-of-the-art (SOTA) performance on the class-imbalanced ZSL datasets AWA1, AWA2, and APY in the Generalized

ZSL setting.

The main contributions of our work are:

1. We propose a novel, simple framework ZSL, where image features from a deep Neural Network are mapped into a latent embedding space to generate latent prototypes for each seen class by a novel triplet training model. Then, a Gaussian Process (GP) regression model is trained by maximizing the marginal likelihood and used to predict latent prototypes of unseen classes.
2. The mapping from image features to a latent space is performed by our proposed triplet training model for ZSL learning, using a novel triplet loss that is robust on ZSL training datasets with highly imbalanced label distribution.
3. Experiment results demonstrate the superior performance of our model on benchmark ZSL datasets like AWA1, AWA2 and APY, as these datasets have imbalanced label distribution on the source domain (training set). Our model has an average training time of 5 minutes on all ZSL datasets, faster than many SOTA models.

Background and Related Works The background about the Gaussian Process and Neural Network models used in this chapter can be found in Chapter 2, Section 2.2.6 and Section 2.2.1 respectively. The related literature on the ZSL problem and Class Imbalance problem are provided in Chapter 2, Section 2.3.5 and Section 2.3.2.

6.2 Proposed Method

6.2.1 Model Overview

In this chapter, we propose a novel method to tackle the label shift correction problem in the ZSL task. Due to the challenging problem setup, we simplify the label shift correction problem to the class imbalance problem in the ZSL task by assuming the target domain has a uniform label distribution over both seen and unseen classes.

Method Summary: We propose a hybrid model for the class imbalance problem in the ZSL task: 1) a Latent Feature Embedding (LFE) model to separate inter-class features that are robust to class-imbalanced datasets, 2) a GP Regression model to predict prototypes of unseen classes based on seen classes and semantic information and 3) a calibrated (bias compensation) classifier to balance the trade-off between seen and unseen class accuracy. A visualization of the model structure is provided in Fig. 5.3.

6.2.2 Latent Feature Embedding Model

Model Structure We propose to learn a linear NN mapping from image features to latent embeddings. We argue that a linear projection with limited flexibility for the ZSL task can help prevent the model from overfitting on seen class training samples. Following others (Dolma and Namboodiri, 2017; Verma and Rai, 2017), we model

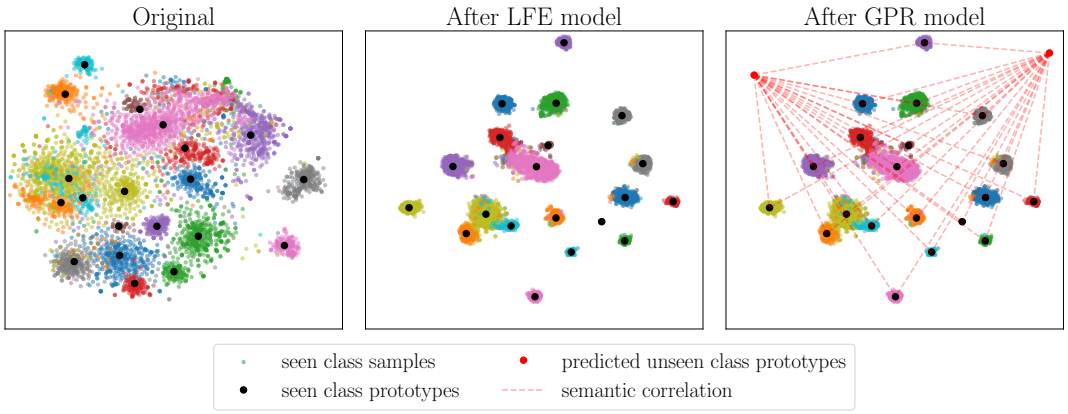

Figure 6.2: Illustration of the proposed ZSL model. Our proposed Latent Feature Embedding (LFE) model helps separate image features in the latent space in a way that is robust to the class-imbalanced training set (source domain) label distribution. Then, we use a Gaussian Process Regression (GPR) model to predict feature prototypes of unseen classes based on feature prototypes of seen classes and semantic correlations between seen and unseen classes. Finally, we use the predicted prototypes the unseen classes and the prototypes of the seen classes to construct a Zero-Shot Learning classifier.

feature vectors from each class using the multivariate Gaussian distribution. We exploit the fact that Gaussian random vectors are closed under linear transformations.

For each feature vector $\boldsymbol{z} \in \mathbb{R}^{N_{feature}}$, the latent embedding $\boldsymbol{x} \in \mathbb{R}^{N_{latent}}$ can be written as:

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{z} + \boldsymbol{b}. \tag{6.1}$$

Here $\boldsymbol{A} \in \mathbb{R}^{N_{latent} \times N_{feature}}$ is a weight parameter matrix and $\boldsymbol{b} \in \mathbb{R}^{N_{latent}}$ is a bias parameter vector.

Triplet Loss Revisited Triplet loss is often used to separate samples from different training classes in the dataset (Huang, Li, Loy et al., 2019b). The standard triplet loss aims to decrease distances between intra-class samples and increase distances between inter-class samples. The triplet loss for a given triplet is defined as

$$\mathcal{L}_T = \max(0, \delta + (x_l - x_m)^2 - (x_l - x_n)^2), \tag{6.2}$$

where x_l, x_m are image samples from the same class, and x_n is the image sample from a different class. The hyperparameter $\delta \in \mathbb{R}^+$ is a positive threshold that balances the inter-class and intra-class distances (Le Cacheux et al., 2019b; Sohn, 2016). In practice, a mini-batch of N triplets are sampled uniformly from the dataset, and the model is then trained.

As seen in Eq. (6.2), the class imbalance problem is not considered in the original triplet loss. Moreover, models trained with a triplet loss usually require many iterations until convergence, expensive memory requirements and a high variance (Sohn, 2016). We thus propose a new Class-Balanced Triplet loss to mitigate these problems.

Class-Balanced Triplet Loss When training a model with a triplet loss, a straightforward approach to tackle the class imbalance problem is to sample class-balanced mini-batch data. If the model is trained using class-balanced data, the class imbalance problem will not affect the performance of the model.

In every iteration, unlike for the traditional triplet loss, we generate a class-balanced mini-batch by sampling $n \in \mathbb{Z}^+$ data points from each one of K training classes in the training set as $\{x_i^{(j)} | j \in \mathcal{Y}, i = 1, 2, ..., n\}$. The batch size is $N = n \times K$. In a supervised classification setting, similar approaches have shown to be effective (Buda et al., 2018a). We then propose a modified triplet loss L_{BT} to train the model on the mini-batch. For every mini-batch, the loss has the form:

$$\mathcal{L}_{BT} = \sum_{i,j} \sum_{l=1}^{n} \max(0, \delta + (x_l^{(i)} - \overline{x^{(i)}})^2 - \min_{m \in \{1,2,..,n\}} (x_m^{(j)} - \overline{x^{(i)}})^2), \tag{6.3}$$

where $\overline{x^{(i)}} := \frac{1}{n} \sum_{m=1}^{n} x_m^{(i)}$ denotes the average of samples from class i in the mini-batch and $\sum_{i,j}$ is the sum over all the different class pairs $i, j \in \mathcal{Y}$.

In Eq. (6.3), replacing the term $x_m^{(i)}$ with $\overline{x^{(i)}}$ in the original triplet loss Eq. (6.2) can help reduce the variance in the loss during training, which is similar to "center loss" (Wen, Zhang et al., 2016). However, unlike their method, we are not adding extra trainable parameters into the model. The $\min(\cdot)$ operation is performed over all samples $x_m^{(j)}$ in class c_j in the mini-batch, which can efficiently reduce computational costs.

With the help of the proposed triplet loss $\mathcal{L}_{BT}$, our model can efficiently learn a latent embedding that separates samples from different classes and performs well on imbalanced datasets.

6.2.3 Gaussian Process Regression Model

We propose a GP Regression model to predict prototypes of unseen classes, leveraging the generalization ability of GP models. Like Mukherjee and Hospedales (2016), we obtain the average of all latent features in each class $\boldsymbol{\mu}^{(i)} = \frac{1}{N^{(i)}} \sum \boldsymbol{x}^{(i)}$ as a prototype for the corresponding class.

We also denote the semantic vector of each class $s^{(i)} \in \mathbb{R}^{N_{semantic}}$. Given the semantic vectors $\boldsymbol{s}^S = [\boldsymbol{s}^{(1)}, ..., \boldsymbol{s}^{(K)}]^T$ and feature prototypes $\boldsymbol{\mu}^S = [\boldsymbol{\mu}^{(1)}, ..., \boldsymbol{\mu}^{(K)}]^T$ for seen class $i \in \mathcal{Y}^S = [1, 2, ..., K]$, along with semantic vectors $\boldsymbol{s}^U = [\boldsymbol{s}^{K+1}, ...\boldsymbol{s}^{K+K'}]^T$ for unseen classes $i \in \mathcal{Y}^U = [K+1, K+2, ..., K+K']$, we can use the GPR model to regress prototypes $\mu^U = [\boldsymbol{\mu}^{K+1}, ...\boldsymbol{\mu}^{K+K'}]^T$ for unseen classes $i \in \mathcal{Y}^U$:

$$\boldsymbol{\mu}^U = f_{GP}(\boldsymbol{s}^U | \theta) + \epsilon. \tag{6.4}$$

Here $f_{GP}(\boldsymbol{s}|\theta)$ is the regression function, $\epsilon \sim \mathcal{N}(0, \sigma^2)$ denotes the Gaussian random noise and θ is the hyperparameter in the model. θ is trained given seen class semantic vectors $\boldsymbol{s}^S$ and corresponding prototypes μ^S.

Directly training a GPR model that learns a projection from $\boldsymbol{s}$ to $\boldsymbol{\mu}$ requires accounting for every dimension in $\boldsymbol{\mu} \in \mathbb{R}^{N_{latent}}$, which is computationally expensive because the model needs to estimate correlations between different dimensions. We propose to avoid this issue by assuming dimensions in $\boldsymbol{\mu}^{(i)}$ are independent over each other so

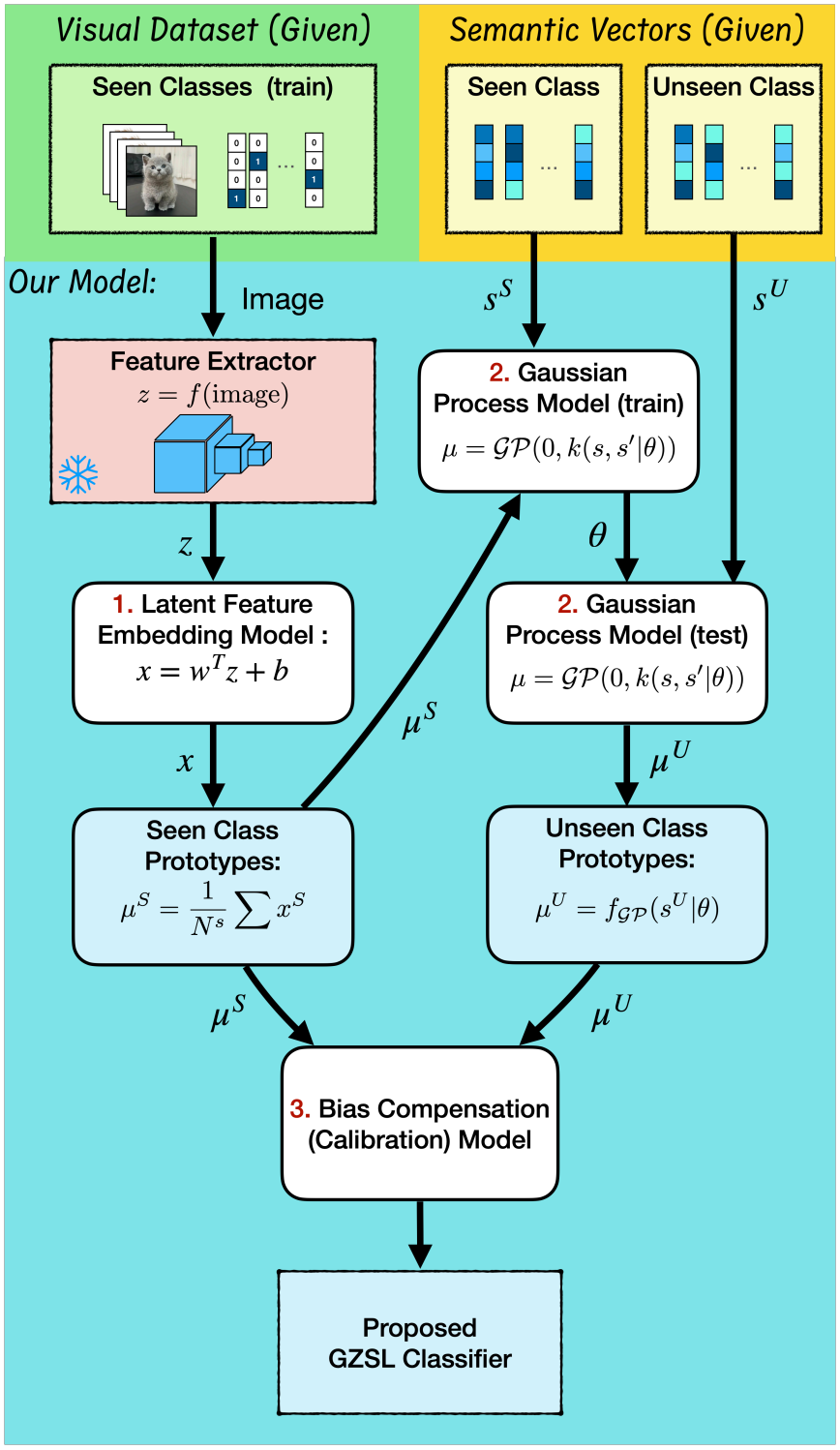

Figure 6.3: Structure of our proposed ZSL model. Feature vectors f are projected to a latent embedding space x, which is trained using proposed Class-Balanced Triplet Loss. A GP model is proposed to predict latent prototypes of unseen classes μ^U, based on latent prototypes of seen class μ^S and semantic vectors from seen and unseen class s^S, s^U.

that the GPR model can be applied to $\boldsymbol{\mu}_i$:

$$\boldsymbol{\mu}_i^U = f_{GP}(\boldsymbol{s}^U|\theta_i) + \epsilon_i. \tag{6.5}$$

Then we have hyperparameter θ_i and noise ϵ_i for i^{th} dimension in μ.

A Gaussian Process is defined by a mean function $m(\boldsymbol{s})$ and a covariance function $k(\boldsymbol{s}, \boldsymbol{s}'|\theta)$ that depends on hyperparameter θ. For $f_{GP}(\boldsymbol{s}|\theta_i)$, a GP can be written as:

$$f_{GP}(\boldsymbol{s}|\theta_i) \sim \mathcal{GP}(\boldsymbol{m}(\boldsymbol{s}), k(\boldsymbol{s}, \boldsymbol{s}'|\theta_i)). \tag{6.6}$$

Here, we will take $\boldsymbol{m} \equiv 0$. The joint prior distribution of seen class prototypes $\boldsymbol{\mu}_i^S$ and regression function $f_{GP}(\boldsymbol{s}|\theta_i)$ can be written as:

$$\begin{bmatrix} f_{GP} \\ \boldsymbol{\mu}_i^S \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} k(\boldsymbol{s}, \boldsymbol{s}|\theta_i) & k(\boldsymbol{s}, \boldsymbol{s}^S|\theta_i) \\ k(\boldsymbol{s}^S, \boldsymbol{s}|\theta_i) & k(\boldsymbol{s}^S, \boldsymbol{s}^S|\theta_i) + \boldsymbol{I}\sigma_i^2 \end{bmatrix}\right), \tag{6.7}$$

where $\boldsymbol{I}$ denotes the identity matrix. $f_{GP} = f_{GP}(\boldsymbol{s}|\theta_i)$ can be obtained via conditioning the joint prior distribution on $\boldsymbol{\mu}_i^S$ to obtain the posterior predictive distribution, which is also a Gaussian distribution $f_{GP}(\boldsymbol{s}|\theta_i)|\boldsymbol{\mu}_i^S, \boldsymbol{s}^S \sim \mathcal{N}(\boldsymbol{m}_i, \Sigma)$. We use the mean $\boldsymbol{m}_i$ of the predictive posterior distribution to form our prediction of $f_{GP}(\boldsymbol{s}|\theta_i)$ evaluated at the unseen classes $\boldsymbol{s} = \boldsymbol{s}^U$, which gives:

$$f_{GP}(\boldsymbol{s}|\theta_i) = \boldsymbol{m}_i(\boldsymbol{s}) = k(\boldsymbol{s}, \boldsymbol{s}^S|\theta_i)\left[k(\boldsymbol{s}^S, \boldsymbol{s}^S|\theta_i) + \boldsymbol{I}\sigma_i^2\right]^{-1}\mu_i^S. \tag{6.8}$$

Any positive semi-definite kernel function may be used as a covariance function $k(\boldsymbol{s}, \boldsymbol{s}'|\theta_i)$, with θ_i as a hyperparameter in the kernel. We propose to search for optimal hyperparameters θ_i for each feature dimension i by maximizing the log marginal likelihoods:

$$\theta_i = \arg\max_{\theta_i}\left(-\frac{1}{2}(\boldsymbol{\mu}_i^S)^T\left[k(\boldsymbol{s}^S, \boldsymbol{s}^S|\theta_i) + \boldsymbol{I}\sigma_i^2\right]^{-1}\boldsymbol{\mu}_i^S - \frac{1}{2}\log k(\boldsymbol{s}^S, \boldsymbol{s}^S|\theta_i)\right). \tag{6.9}$$

With θ_i given, unseen class prototypes can be evaluate by $\mu_i^U = f_{GP}(\boldsymbol{s}|\theta_i)$. Like other prototypical methods, the classifier can be constructed using a nearest-neighbour approach based on a distance metric. We use the Euclidean distance in our model:

$$predict(\boldsymbol{x}) = \arg\min_{i \in \mathcal{Y}^S \cup \mathcal{Y}^U} \|\boldsymbol{x} - \mu^{(i)}\|^2. \tag{6.10}$$

6.2.4 Calibration (Bias Compensation) Model

It is well known that ZSL models trained on seen classes are biased towards classifying unseen images into seen classes (Le Cacheux et al., 2019a; Skorokhodov and Elhoseiny, 2021). Therefore, it is necessary to add a penalty term $\gamma \in \mathbb{R}^+$ when computing classification metrics over seen classes $i \in \mathcal{Y}^S$. We adopt the calibration

approach proposed by Le Cacheux et al. (2019a)[1]. The calibrated nearest neighbor classifier is then written as:

$$predict(\boldsymbol{x}) = \underset{i \in \mathcal{Y}^S \cup \mathcal{Y}^U}{\arg\max} [-\|\boldsymbol{x} - \boldsymbol{\mu}^{(i)}\|^2 - \gamma \mathbb{I}_{\in \mathcal{Y}^S}(i)]. \tag{6.11}$$

where $\mathbb{I}_{i \in \mathcal{Y}^S}$ is the indicator function, which equals 1 when class i is from a seen class $\mathcal{Y}^S$ and 0 otherwise.

In our model, we first use the GPR model to predict the validation class based on the training class; then, we train γ as a calibration penalty to maximize the harmonic mean. After training, the test class is predicted and conditioned on training classes and validation classes together, and γ is used for calibration evaluation of the performance on the test set.

6.3 Experiments

6.3.1 Experimental Setup

Datasets: We test the performance of our model on five benchmark datasets, namely: Animals with Attributes 1 (AWA1) (Xian, Lampert et al., 2019), Animals with Attributes 2 (AWA2) (Xian, Lampert et al., 2019), A Pascal and A Yahoo (APY) (Farhadi et al., 2009), Caltech UCSD Birds 200-2011 (CUB) (Wah et al., 2011) and SUN Attribute (SUN) (Patterson and Hays, 2012). Detailed information on the datasets used is provided in Table 6.1 below.

We also provide the total, average, maximum and minimum sample number per class in Tab. 6.1. We can see that CUB and SUN are relatively class-balanced datasets because of a low variance in the number of samples per class. While AWA2, AWA1 and APY are class-imbalanced datasets. In particular, for APY, one class contains $1/3$ of the total number of samples in the dataset.

Dataset		CUB	SUN	AWA2	AWA1	APY
Class	seen	150	645	40	40	20
	unseen	50	72	10	10	12
Feature Dim		2048	2048	2048	2048	2048
Attribute Dim		312	102	85	85	64
Total sample No.		11788	14340	37322	30475	15339
Average Sample No. per-class		58	20	746	609	479
Max Sample No. per-class		60	20	1645	1168	5071
Min Sample No. per-class		41	20	100	92	51

Table 6.1: Zero-Shot Learning Datasets Information. AWA1, AWA2 and APY are class-imbalanced datasets.

[1]Note that the term "calibration" here in the ZSL setting is different to that used in the uncertainty quantification literature (Guo, Pleiss et al., 2017)

"Proposed Split V2.0": In the ZSL setting, datasets are usually divided into sets of seen classes and sets of unseen classes. Most recent models adopt "Proposed Split" proposed by Xian, Lampert et al. (2019) to test the performance of their model. In September 2020, Xian, Lampert et al. (2019) updated their paper with "Proposed Split V2.0" to fix the problem that the old "Proposed Split" has testing seen class samples included in the training samples. Such issues may have a significant impact on the performance current SOTA models. In this chapter, we report the performance of previous models reproduced on "Proposed Split V2.0" by other papers and our own to ensure a fair comparison.

Implementation Detail: Our model is implemented using PyTorch (Paszke et al., 2017) and GPytorch (Gardner et al., 2018) and trained on an NVIDIA RTX 2080Ti GPU machine. We use feature vectors extracted by a pre-trained ResNet101 network, proposed by Xian, Lampert et al. (2019). As argued by Le Cacheux et al. (2019b), the feature vector space is unbounded, and a few feature vectors with high values may hinder the network learning from the triplet loss. In this chapter, we preprocess the feature vectors by clipping the features by 7 and scaling to range $[0, 1]$. The neural network model is trained with the Adam (Kingma and Ba, 2015) optimizer with a learning rate 0.002 and weight decay 0.1 for 500 episodes on each ZSL dataset. We set the threshold $\delta = 4$ in our triplet loss. The GPR model is also trained with the Adam (Kingma and Ba, 2015) optimizer with a learning rate of 0.01 for 1000 epochs for each ZSL dataset. Details of the hyperparameter search can be found in the Ablation Study section.

6.3.2 State-of-the-art Comparison

We compare the performance of our model with several state-of-the-art (SOTA) ZSL models in Tab. 6.3. A_T refers to Traditional ZSL per-class Top-1 Accuracy. A_U, A_S refers to Generalized ZSL unseen and seen class Top-1 per-class Accuracy respectively. Harmonic mean $H = 2(A_U * A_S)/(A_U + A_S)$ measures the trade-off between seen and unseen class accuracy.

The reported performance of SYNC (Changpinyo et al., 2016), ALE (Akata et al., 2015), DEVISE (Frome et al., 2013), GFZSL (Verma and Rai, 2017) are updated by Xian, Lampert et al. (2019). GDAN (Huang, Wang, Yu et al., 2019), CADA-VAE (Schönfeld et al., 2019), TF-VAEGAN (Narayan et al., 2020), LisGAN (Li, Jing et al., 2019), GCM-CF (Yue, Wang et al., 2021) were updated by GCM-CF (Yue, Wang et al., 2021). FREE (Chen, Wang et al., 2021) and CNZSL (Skorokhodov and Elhoseiny, 2021) adopt "Proposed Split V2.0" already in their paper. Performance of E-PGN (Yu et al., 2020), Li, Min et al. (2019) and DVBE (Min et al., 2020) on "Proposed Split V2.0" are fintuned and updated by the author using the published official code of each paper.

Following Yue, Wang et al. (2021), we have not listed models that only report performance on incorrect "Proposed Split", including f-VAEGAN-D2 (Xian, Sharma et al., 2019), RELATION NET (Sung et al., 2018), DAZLE (Huynh and Elhamifar, 2020), OCD (Keshari et al., 2020), IZF (Shen et al., 2020), AGZSL (Chou, Lin et al., 2020), IPN (Liu, Zhou et al., 2020) and CE-GZSL (Han, Fu, Chen et al., 2021). SOTA models that only report ImageNet performance like DGP (Kampffmeyer et al.,

Methods	Provided by	AWA2				AWA1				APY			
		ZSL	GZSL			ZSL	GZSL			ZSL	GZSL		
		A_T	A_U	A_S	H	A_T	A_U	A_S	H	A_T	A_U	A_S	H
Class-Imbalanced Datasets													
SYNC	Xian *et al.*	49.3	9.7	**89.7**	17.5	51.8	9.0	**88.9**	16.3	23.9	7.4	66.3	13.3
ALE	Xian *et al.*	62.5	14.0	81.8	23.9	59.9	16.8	76.1	27.5	39.7	4.6	73.7	8.7
DEVISE	Xian *et al.*	59.7	17.1	74.7	27.8	54.2	13.4	68.7	22.4	37.0	3.5	78.4	6.7
GFZSL	Xian *et al.*	63.8	2.5	80.1	4.8	68.2	1.8	80.3	3.5	38.4	0.0	**83.3**	0.0
GDAN	CFZSL	-	26.0	78.5	39.1	-	-	-	-	-	29.0	63.7	39.9
CADA-VAE	CFZSL	-	55.4	76.1	64.0	-	-	-	-	-	34.0	54.2	41.7
TF-VAEGAN	CFZSL	-	52.5	82.4	64.1	-	-	-	-	-	31.7	61.5	41.8
LisGAN	CFZSL	-	53.1	68.8	60.0	-	-	-	-	-	33.2	56.9	41.9
Li *et al.*	author	-	-	-	-	69.4	59.2	78.4	67.5	-	-	-	-
E-PGN	author	67.4	32.1	66.6	43.3	**71.1**	56.8	81.2	66.9	-	-	-	-
DVBE	author	-	45.4	76.9	57.1	-	-	-	-	-	32.9	47.6	38.9
GCM-CF	CFZSL	-	60.4	75.1	67.0	-	-	-	-	-	37.1	56.8	44.9
CNZSL	CFZSL	-	60.2	77.1	67.6	-	63.1	73.4	67.8	-	-	-	-
FREE	FREE	-	60.4	75.4	67.1	-	62.9	69.4	66.0	-	-	-	-
$\mathcal{L}_{BT}$ + GP(**ours**)	author	**68.6**	**62.2**	76.7	**68.7**	70.1	**64.5**	73.3	**68.6**	**47.1**	**42.8**	64.3	**51.4**

Table 6.2: Performance of Zero-Shot Learning Models comparison on Class-Imbalanced datasets (label shift). ZSL Top-1 per-class Accuracy on "Proposed Split V2.0", Traditional ZSL as A_T, Generalized unseen, seen and harmonic mean as A_U, A_S, H respectively. Our model outperforms previous models on class-imbalanced datasets AWA2, AWA1 and APY.

2019) and HVE (Liu, Chen et al., 2020), or only report transductive ZSL results like SDGN (Wu, Zhang et al., 2020) are also not listed. A detailed discussion can be found in the supplementary material.

As seen from Tab. 6.3,6.2, our model has reached SOTA performance on the AWA2, AWA1 and APY datasets. Our model outperforms SOTA results by a large margin, especially on the APY dataset, where the dataset has a significant class-imbalance data distribution. Our model has a somewhat lower performance on the CUB and SUN datasets. This may be because CUB and SUN are fine-grained datasets, and our latent embedding network cannot efficiently capture minor differences between classes in these datasets.

6.3.3 Training Speed Comparison

The average training times of SOTA models on each dataset are reported in Tab. 6.4. With the help of our adjusted triplet loss, our model can be trained within as little as a few minutes on all ZSL datasets. The only model that trains faster than ours is CNZSL (Skorokhodov and Elhoseiny, 2021). However, our model performs better than theirs on all ZSL datasets except SUN. GDAN, E-PGN, and CADA-VAE have training times similar to our model but with lower performance on the GZSL task.

6.3.4 Area Under Seen and Unseen Curve (AUSUC)

For the GZSL problem, models usually have to balance the trade-off between seen and unseen class accuracies, which is measured by the Harmonic mean H. Similar to our model, many SOTA models like (Skorokhodov and Elhoseiny, 2021; Yue, Wang et al., 2021) introduced a calibration parameter γ to account for the trade-off.

Methods	Provided by	CUB				SUN			
		ZSL	GZSL			ZSL	GZSL		
		A_T	A_U	A_S	H	A_T	A_U	A_S	H
Class-Balanced Datasets									
SYNC	Xian *et al.*	56.0	11.5	70.9	19.8	56.2	7.9	43.3	13.4
ALE	Xian *et al.*	54.9	23.7	62.8	34.4	58.1	21.8	33.1	26.3
DEVISE	Xian *et al.*	52.0	23.8	53.0	32.8	56.5	16.9	27.4	20.9
GFZSL	Xian *et al.*	49.3	0.0	45.7	0.0	60.8	0.0	39.6	0.0
GDAN	CFZSL	-	35.0	28.7	31.6	-	38.2	19.8	26.1
CADA-VAE	CFZSL	-	50.3	56.1	53.0	-	43.6	36.4	39.7
TF-VAEGAN	CFZSL	-	50.7	**62.5**	56.0	-	41.0	39.1	41.0
LisGAN	CFZSL	-	44.9	59.3	51.1	-	41.9	37.8	39.8
E-PGN	author	**69.1**	50.1	60.0	54.6	-	-	-	-
DVBE	author	-	46.7	51.4	48.9	-	34.7	32.3	33.4
GCM-CF	CFZSL	-	**61.0**	59.7	**60.3**	-	47.9	37.8	42.2
CNZSL	CFZSL	-	49.9	50.7	50.3	-	44.7	**41.6**	**43.1**
FREE	FREE	-	55.7	59.9	57.7	-	47.4	37.2	41.7
$\mathcal{L}_{BT}$ + GP(**ours**)	author	59.9	50.1	56.3	53.1	**63.2**	**50.4**	34.8	41.2

Table 6.3: ZSL Top-1 per-class Accuracy on "Proposed Split V2.0", Traditional ZSL as A_T, Generalized unseen, seen and harmonic mean as A_U, A_S, H respectively. Our model is comparable with SOTA models on the Class-Balanced datasets

Dataset	CUB	SUN	AWA2	AWA1	APY
GDAN (Huang, Wang, Yu et al., 2019)	8min	18min	14min	-	7min
CADA-VAE (Schönfeld et al., 2019)	3min	5min	6min	6min	-
E-PGN (Yu et al., 2020)	5min	-	9min	8min	-
DVBE (Min et al., 2020)	180min	-	540min	-	210min
CNZSL (Skorokhodov and Elhoseiny, 2021)	0.5min	0.5min	0.5min	0.5min	-
$\mathcal{L}_{BT}$ + GP(**ours**)	5min	8min	3min	3min	2min

Table 6.4: Average Training Time (minutes) for different models with an NVIDIA RTX 2080 Ti GPU card on each ZSL dataset. The training time of our model is competitive with other models

Datasets	CUB				SUN				AWA2				AWA1				APY			
	ZSL	GZSL			ZSL	GZSL			ZSL	GZSL			ZSL	GZSL			ZSL	GZSL		
	A_T	A_U	A_S	H	A_T	A_U	A_S	H	A_T	A_U	A_S	H	A_T	A_U	A_S	H	A_T	A_U	A_S	H
KRR	20.8	14.6	24.1	18.8	40.0	29.8	19.3	23.4	43.9	28.9	61.2	39.3	43.3	28.6	62.8	39.3	34.7	26.4	70.1	38.4
$\mathcal{L}_{BT}$ + KRR	22.1	16.4	25.5	20.0	40.1	28.5	23.0	25.5	44.2	32.9	54.4	41.0	43.7	34.8	55.6	42.8	35.3	31.3	63.9	42.0
GP	53.3	42.4	46.3	44.2	61.9	51.7	32.9	40.2	**69.2**	54.7	**78.0**	64.3	69.7	57.6	**73.9**	64.7	38.3	31.5	72.6	44.0
$\mathcal{L}_T$+ GP	57.0	48.1	51.3	49.7	57.4	48.3	25.7	33.5	64.5	56.7	75.8	64.9	66.6	59.7	73.3	65.8	40.5	34.1	**75.2**	47.0
$\mathcal{L}_{BT}$ + GP(**ours**)	**59.9**	**50.1**	**56.3**	**53.1**	**63.2**	**50.4**	**34.8**	**41.2**	68.6	**62.2**	76.7	**68.7**	**70.1**	**64.5**	73.3	**68.6**	**47.1**	**42.8**	64.3	**51.4**

Table 6.5: Ablation Study on different model structures. Our proposed $\mathcal{L}_{BT}$ + GP model performs consistently better than Kernel Ridge Regression (KRR) models, the Gaussian Process (GP) model and the GP model with the original triplet loss $\mathcal{L}_T$ + GP.

Recently, Yue, Wang et al. (2021) proposed to utilize γ and plot the Area Under the Seen and Unseen Curve (AUSUC). Such a Fig. can provide a more detailed measure of the seen and unseen class trade-off. We compare the AUSUC curve of our model with CADA-VAE (Schönfeld et al., 2019) and CNZSL (Skorokhodov and Elhoseiny, 2021), which have official codes available.

From Fig. 6.4, we can see that our model performs consistently better than CNZSL

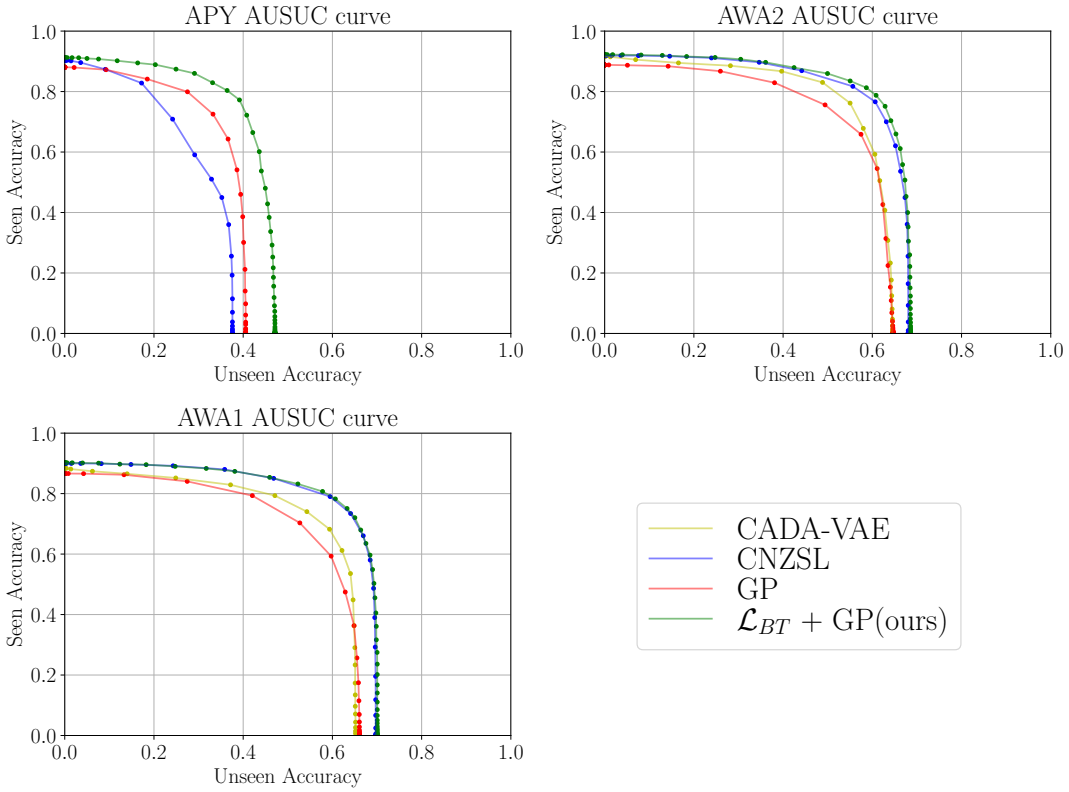

Figure 6.4: Comparison of Area Under Seen and Unseen Curve (AUSUC) for each dataset with CNZSL (Skorokhodov and Elhoseiny, 2021) and CADA-VAE (Schönfeld et al., 2019). Our model is performing consistently better than CNZSL and CADA-VAE on class-imbalanced datasets AWA2, AWA1 and APY.

class	cow	horse	motorbike	person	pottedplant	sheep	train	tvmonitor	donkey	goat	jetski	statue
Sample Frequency	2.48%	3.81%	3.75%	63.99%	5.50%	2.95%	2.22%	3.77%	1.75%	2.06%	5.04%	2.61%
$\mathcal{L}_T$ + GP	7.1	**36.9**	75.7	7.4	31.4	17.9	80.1	75.9	12.2	**63.2**	**51.4**	13.0
$\mathcal{L}_{BT}$ + GP**(ours)**	**15.7**	35.9	**75.8**	**12.3**	**33.9**	**19.2**	**84.6**	**82.3**	**36.7**	48.5	40.6	**28.5**

Table 6.6: Ablation Study of per-class accuracy for unseen classes on class-imbalanced dataset APY. Our proposed $\mathcal{L}_{BT}$ + GP model performs better than the GP model with the original triplet loss $\mathcal{L}_T$ + GP consistently on most of the classes.

and CADA-VAE on class-imbalanced datasets AWA2 and APY and is competitive with CNZSL on the AWA1 dataset.

6.3.5 Ablation Study

Model Structure Ablation: We compare the performance of our model with several similar models that have different model structures in Tab. 6.5. We report the performance of the Kernel Ridge Regression (KRR) model on feature space (as a baseline for GPR), KRR on latent space which is trained with our proposed $\mathcal{L}_{BT}$ loss, the Gaussian Process (GP) model on feature space and the GP model on latent space trained with the original triplet loss $\mathcal{L}_T$. Our model performs consistently better than all the other baseline approaches on each ZSL dataset.

Hyperparameter Ablation: We analyze the influence of two main hyperparameters on the performance of our model. These hyperparameters are the clip value used for preprocessing feature vectors and the threshold δ used in triplet loss. As seen from Fig. 6.5, our model is not sensitive to the clip number used in data preprocessing when the clip number is in the range $[4, 9]$. As long as $\delta > 3$, the performance of our model on each dataset is relatively stable.

Per-Class Accuracy Ablation: In Tab. 6.6, we also provide per-class accuracy of our model $\mathcal{L}_{BT} + GP$ and traditional triplet loss $\mathcal{L}_T$ + GP, on unseen classes on the APY dataset. As can be seen from the table, the APY dataset has a highly class-imbalanced unseen class distribution, where 64% of the unseen test samples come from the class "person". Our model performs consistently better than traditional triplet loss models in most classes.

Comparison with Recent Models: A variety of models have been proposed in recent years for more advanced ZSL tasks (Hong, Hayder et al., 2023; Zheng et al., 2023). Compared with these models, our propose model only inference image prototypes of unseen classes based on semantic embeddings, while recent models are usually empowered by foundation models like CLIP (Radford et al., 2021) or utilize more advanced NN training paradigm like federate learning (Asif et al., 2024).

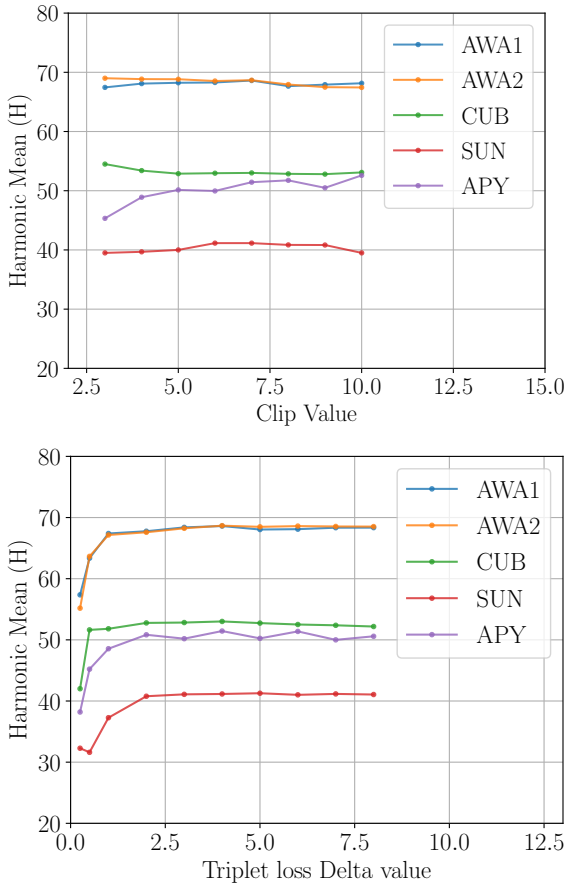

Figure 6.5: Harmonic mean H for each dataset influenced by the hyperparameter Clip value in preprocessing (left) and δ in triplet loss (right). Our model is stable w.r.t. these hyperparameters

6.4 Conclusion

In this chapter, we propose a novel model that combines a Neural Network and Gaussian Process regression to tackle the label shift problem in the ZSL and Generalized ZSL task. We propose an NN model that projects feature vectors into a latent embedding space and generates latent prototypes of seen classes. A GP model is then trained to predict prototypes of unseen classes. Finally, a ZSL classifier is constructed using the prototypes.

We trained our NN model with a Class-Balanced Triplet loss that mitigates the problem of class imbalance in ZSL datasets. Experiments demonstrate that our model, though employing a simple design, can reach SOTA performance on the class-imbalanced ZSL datasets AWA1, AWA2 and APY in the Generalized ZSL setting.

Chapter 7

Conclusions

This thesis mainly addresses the problem of label shift in the closed set classification, open set classification and Zero-Shot classification tasks. The objectives of the thesis are:

- **Chapter 3**: Design a classifier-based closed set label shift model that is robust under large label shift.
- **Chapter 4**: Design a feature-based closed set label shift model that is feasible for different types of classifiers.
- **Chapter 5**: Design a open set label shift model that could utilize existing OOD detection models without re-training or finetuning.
- **Chapter 6**: Design zero-shot learning model that is robust under imbalanced source label distribution.

7.1 Summary

The main contributions of this thesis are:

1. A novel classifier based label shift estimation model for the closed set image classification task (Chapter 3). The model includes 1) an EM algorithm that is guaranteed to converge to the unique MAP estimate of the target label distribution when the prior distribution is a Dirichlet distribution and 2) an Adaptive Prior Learning model that determines the prior parameter given available data. The i.i.d. samples from the Bayesian posterior of the target label distribution can be sampled with an MCMC algorithm when the uncertainty of the estimation is important. Experimental results demonstrate that the proposed label shift estimation model is robust even when the source or target domain has a highly imbalanced label distribution.
2. A novel feature based label shift estimation framework – GLSE framework, for the closed set image classification task (Chapter 4). The framework includes 1) a statistical inference model that estimates the parameters of the conditional distribution of feature given label for different feature extractors and 2) a latent variable model with two EM algorithms, which converge to an MLE and the unique MAP estimate of the target label distribution respectively. Four label shift estimation models are proposed under the framework for different

types of NN classifiers. Moreover, the MLLS model proposed by previous work (Saerens et al., 2002) is shown to be a special case of the proposed framework. Experimental results justify the robustness and effectiveness of the proposed models.

3. A novel classifier based label shift estimation model for the open set image classification task (Chapter 5). With a reference OOD dataset at the test time, the model is constructed with 1) a statistical inference model that estimates the source ID/OOD data ratio, 2) an EM algorithm that is guaranteed to converge to a MLE of the target label distribution for ID classes and the OOD class and 3) a statistical inference model that estimate target ID/OOD ratio when assumption on ID/OOD classifier in 1) and 2) is not satisfied. The sampling errors of estimates in 1) and 3) are quantified with a concentration inequality. Experimental results justify the effectiveness of the model with a pre-trained ID classifier and different OOD classifiers proposed in previous OOD detection literature.

4. A novel Zero-Shot Learning model that is robust when the source domain (train) dataset has a highly imbalanced label distribution (Chapter 6). The model consists of 1) a Neural Network feature extractor trained with a novel Class-Balanced Triplet loss to equally cluster image samples for each seen class in the feature space, 2) a Gaussian Process Regression model that predicts the prototypes of the unseen classes and 3) a prototype classifier constructed based on prototypes of seen and unseen classes.

To facilitate practical applications, this thesis provides a comparison between existing popular closed set and open set label shift models and our models in Tab. 7.1. As shown in the table, the proposed model requires no retraining/finetuning of the classifier, at the cost of some extra assumptions on the source domain classifier. The validity of these extra assumptions are discussed in the respective chapters – Assumption 3 for MAPLS model in Chapter 3, Assumption 4 for GLSE models in Chapter 4 and Assumption 5 for OSLS-MLE/MAP model in Chapter 5.

Model	MLLS	MAPLS (**ours**)	BBSE/RLLS/IWGAN	GLSE (**ours**)	PULSE	OSLS (**ours**)
Extra Assumption	✓	✓	✗	✓(✗ for GLSE_{Cat})	✗	✓
Close Set Estimation	✓	✓	✓	✓	✓	✓
Open Set Estimation	✗	✗	✗	✗	✓	✓
NO (re)training	✓	✓	✓	✓	✗	✓
Allow Prior Information	✗	✓	✗	✓	✗	✓

Table 7.1: Difference between our model and other Closed/Open Set Label Shift estimation and correction models. BBSE/RLLS/IWGAN and PULSE are effective without extra assumptions on the classifier required under their corresponding label shift problem setup. However, if the corresponding assumptions are roughly satisfied, MLLS and our MAPLS/GLSE/OSLS-MAP/MLE models could be more effective, especially when prior information is available or re-training/finetuning is expensive.

In practice, BBSE/RLLS/IWGAN and PULSE models are effective without any extra assumptions on the classifier. However, these models could suffer from high estimation error under large label shift or with limited samples (*e.g.* Tab. 3.5). On the

other hand, if the corresponding assumptions are roughly satisfied – which can be justified through calibration performance metrics like ECE (Alexandari et al., 2020), MLLS and our MAPLS/GLSE/OSLS-MAP/MLE models could be more effective. A collection of the implementations of MLLS, BBSE, RLLS, MAPLS methods can be found in our official code of the MAPLS model at https://github.com/ChangkunYe/MAPLS. Our OSLS model implementation is publicly available at https://github.com/ChangkunYe/OpenSetLabelShift.

7.2 Limitations

The proposed models have several limitations that restrict their applicability, where the most significant limitations include the following:

1. **Assumptions on Classifier**: The proposed closed set and Open Set Label Shift estimation model are established based on the assumptions of the classifiers or feature extractors. For example, the label shift estimation model proposed in Chapter 3 assumes the classifier f models the conditional probability $p_s(y|x)$ (Assumption 3). Although our experiments show that the proposed models are empirically effective, whether these assumptions can be relaxed can be further discussed in future works.

2. **Convergence Rate:** Generally, EM algorithms can suffer from the problem of slow convergence near the optimal point (Varadhan and Roland, 2008). Therefore, the EM algorithm derived in the proposed label shift estimation models in Chapter 3, 4, and 5 can also suffer from a slow convergence rate. Although experiments demonstrate that the proposed MAPLS model (Chapter 3) for the CSLS problem is able to converge in less than 100 iterations, more theoretical analysis can be provided to justify these empirical results.

3. **Consistency Analysis:** This thesis does not discuss the consistency of the proposed estimation model, *i.e.* whether the estimator is able to converge to the ground truth value when sufficient data is available. Although consistency analysis on the similar approach MLLS (Saerens et al., 2002) yields mild requirement on the classifier f (Garg, Wu, Balakrishnan et al., 2020), whether these conditions can be extended to the proposed models requires further study.

4. **Model Robustness on Open Set:** Whether an Open Set Label Shift estimation model is robust when the source or target domain has a highly imbalanced label distribution is rarely studied. The Open Set Label Shift setting is more realistic but challenging than the Closed Set Label Shift. The proposed Open Set Label Shift estimation model has special requirements for the OOD detection classifier, which may not be satisfied when the source domain has a highly imbalanced label distribution.

7.3 Ongoing Future Works

Several theoretical and practical areas can be further investigated with the help of the results of this thesis.

1. **Error Bound**: For the label shift estimation models proposed in Chapter 3, 4, 5, it is always good to know the upper bound of the estimation error of the proposed model. The estimation error bound for the MLLS model (Saerens et al., 2002) was provided by Garg, Wu, Balakrishnan et al. (2020), which could be potentially extended for our MAPLS algorithm (Chapter 3) and GLSE algorithms (Chapter 4) in future research. The error bound of the OSLS-MLE/MAP algorithm (Chapter 5) for the open set problem needs further investigation due to the challenging open set problem setup.

2. **Assumptions Relaxation**: The label shift estimation models proposed in Chapter 3, 4, 5 are usually established on the assumption that the classifier f reflects the ground truth conditional probability $p(y|x)$. Practical classifiers may not satisfy this kind of assumption. Therefore, it is worth exploring the relaxed assumption of the label shift estimation model.

3. **Model Calibration**: As shown by Garg, Wu, Balakrishnan et al. (2020), similar to MLLS, the assumption $p_s(y|x) = f(x)$ in our MAPLS model (Chapter 3) can be replaced with the canonical calibration condition $p_s(y|f(x)) = f(x)$. This implies that a classifier with better calibration performance will better suit our proposed label shift estimation model. Therefore, the label shift estimation model can be potentially used to analyze the calibration performance of a classifier.

4. **Label Shift Correction**: This thesis mainly focuses on the label shift estimation problem. The ultimate objective of the label shift problem, however, is to estimate and correct label shift and provide a target domain classifier. Therefore, future research could be conducted on label shift correction based on the robust label shift estimation model proposed in this thesis.

5. **Online Label Shift**: Label shift with a target domain data stream is also a practical problem setup. Existing approaches like Wu, Guo, Su et al. (2021); Zhao et al. (2021) are based on the previous static label shift estimation models like BBSE, which may not be robust under significant label shift (Chapter 3). Therefore, our robust label shift estimation model MAPLS and GLSE can be used to solve these problems and improve estimation performance.

6. **Joint Distribution Shift**: A practical classification problem can face a mixture of label shift, covariate shift and other types of distribution shift. It is worth developing a distribution shift correction model that tackles the most common types of shift with the help of our proposed label shift estimation model.

Bibliography

Akata, Z., Perronnin, F., Harchaoui, Z. and Schmid, C. (2015). 'Label-embedding for image classification'. *IEEE transactions on pattern analysis and machine intelligence*, 38(7), pp. 1425–1438 (cited on pp. 33, 115).

Alexandari, A., Kundaje, A. and Shrikumar, A. (2020). 'Maximum likelihood with bias-corrected calibration is hard-to-beat at label shift adaptation'. In: *International Conference on Machine Learning*. PMLR, pp. 222–232 (cited on pp. 2, 4, 12, 26, 29, 38, 46–48, 52, 55, 68–70, 87, 90, 123, 144, 148, 149, 161, 169, 181).

Ali, A., Shamsuddin, S. M. and Ralescu, A. L. (2013). 'Classification with class imbalance problem'. *Int. J. Advance Soft Compu. Appl*, 5(3), pp. 176–204 (cited on p. 30).

Asif, M., Naz, S., Ali, F., Salam, A., Amin, F., Ullah, F. and Alabrah, A. (2024). 'Advanced zero-shot learning (AZSL) framework for secure model generalization in federated learning'. *IEEE Access* (cited on p. 119).

Azizzadenesheli, K., Liu, A., Yang, F. and Anandkumar, A. (2018). 'Regularized Learning for Domain Adaptation under Label Shifts'. In: *International Conference on Learning Representations* (cited on pp. 25, 27, 29, 48, 55, 70, 91, 149, 161, 180).

Baby, D., Garg, S., Yen, T.-C., Balakrishnan, S., Lipton, Z. and Wang, Y.-X. (2024). 'Online label shift: Optimal dynamic regret meets practical algorithms'. *Advances in Neural Information Processing Systems*, 36 (cited on pp. 27–29).

Barandela, R., Sánchez, J. S., Garcıa, V. and Rangel, E. (2003). 'Strategies for learning in class imbalance problems'. *Pattern Recognition*, 36(3), pp. 849–851 (cited on p. 30).

Bele, P., Bundele, V., Bhattacharya, A., Jha, A., Roig, G. and Banerjee, B. (2024). 'Learning Class and Domain Augmentations for Single-Source Open-Domain Generalization'. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1816–1826 (cited on p. 3).

Bendale, A. and Boult, T. E. (2016). 'Towards open set deep networks'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1563–1572 (cited on pp. 32, 82, 91, 179).

Betancourt, M. (2017). 'A conceptual introduction to Hamiltonian Monte Carlo'. *arXiv preprint arXiv:1701.02434* (cited on pp. 20, 45).

Bezdek, J. C. and Kuncheva, L. I. (2001). 'Nearest prototype classifier designs: An experimental study'. *International journal of Intelligent systems*, 16(12), pp. 1445–1473 (cited on p. 55).

Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P., Horsfall, P. and Goodman, N. D. (2018). 'Pyro: Deep Universal Probabilistic Programming'. *Journal of Machine Learning Research* (cited on pp. 47, 70).

Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*. Vol. 4. 4. Springer (cited on pp. 18, 21).

Bitterwolf, J., Mueller, M. and Hein, M. (2023). 'In or Out? Fixing ImageNet Out-of-Distribution Detection Evaluation'. In: *ICML*. (cited on p. 90).

Boney, R. and Ilin, A. (2017). 'Semi-supervised few-shot learning with prototypical networks'. *CoRR abs/1711.10856* (cited on pp. 31, 33).

Box, G. E. and Tiao, G. C. (2011). *Bayesian inference in statistical analysis*. John Wiley & Sons (cited on pp. 17–19).

Brooks, S. (1998). 'Markov chain Monte Carlo method and its application'. *Journal of the royal statistical society: series D (the Statistician)*, 47(1), pp. 69–100 (cited on p. 19).

Buda, M., Maki, A. and Mazurowski, M. A. (2018a). 'A systematic study of the class imbalance problem in convolutional neural networks'. *Neural Networks*, 106, pp. 249–259 (cited on pp. 1, 30, 108, 111).

Buda, M., Maki, A. and Mazurowski, M. A. (2018b). 'A systematic study of the class imbalance problem in convolutional neural networks'. *Neural Networks*, 106, pp. 249–259 (cited on pp. 3, 5, 30).

Byrd, J. and Lipton, Z. (2019). 'What is the effect of importance weighting in deep learning?' In: *International Conference on Machine Learning*. PMLR, pp. 872–881 (cited on p. 30).

Cai, J., Wang, Y. and Hwang, J.-N. (2021). 'Ace: Ally complementary experts for solving long-tailed recognition in one-shot'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 112–121 (cited on p. 31).

Cao, K., Wei, C., Gaidon, A., Arechiga, N. and Ma, T. (2019). 'Learning imbalanced datasets with label-distribution-aware margin loss'. *Advances in neural information processing systems*, 32 (cited on pp. 30, 31, 44, 47, 69).

Caruana, R. (1996). 'Algorithms and applications for multitask learning'. In: *ICML*. Citeseer, pp. 87–95 (cited on p. 3).

Casella, G. (1992). 'Illustrating empirical Bayes methods'. *Chemometrics and intelligent laboratory systems*, 16(2), pp. 107–125 (cited on p. 42).

Chan, Y. S. and Ng, H. T. (2005). 'Word Sense Disambiguation with Distribution Estimation.' In: *IJCAI*. Vol. 5, pp. 1010–5 (cited on p. 4).

Changpinyo, S., Chao, W.-L., Gong, B. and Sha, F. (2016). 'Synthesized classifiers for zero-shot learning'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5327–5336 (cited on pp. 107, 115).

Chapelle, O., Scholkopf, B. and A. Zien, E. (2009). 'Semi-Supervised Learning (Chapelle, O. et al., Eds.; 2006) [Book reviews]'. 20(3), pp. 542–542 (cited on p. 33).

Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P. (2002). 'SMOTE: synthetic minority over-sampling technique'. *Journal of artificial intelligence research*, 16, pp. 321–357 (cited on p. 30).

Chen, G., Peng, P., Wang, X. and Tian, Y. (2021). 'Adversarial reciprocal points learning for open set recognition'. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11), pp. 8065–8081 (cited on pp. 5, 32).

Chen, S., Wang, W., Xia, B., Peng, Q., You, X., Zheng, F. and Shao, L. (2021). 'Free: Feature refinement for generalized zero-shot learning'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 122–131 (cited on pp. 107, 115).

Chou, H.-P., Chang, S.-C., Pan, J.-Y., Wei, W. and Juan, D.-C. (2020). 'Remix: Rebalanced mixup'. In: *European Conference on Computer Vision*. Springer, pp. 95–110 (cited on p. 31).

Chou, Y.-Y., Lin, H.-T. and Liu, T.-L. (2020). 'Adaptive and generative zero-shot learning'. In: *International Conference on Learning Representations* (cited on pp. 115, 186).

Cui, Y., Jia, M., Lin, T.-Y., Song, Y. and Belongie, S. (2019). 'Class-Balanced Loss Based on Effective Number of Samples'. In: *CVPR* (cited on p. 108).

Djurisic, A., Bozanic, N., Ashok, A. and Liu, R. (2022). 'Extremely simple activation shaping for out-of-distribution detection'. *arXiv preprint arXiv:2209.09858* (cited on pp. 32, 91, 179).

Dolma, Y. and Namboodiri, V. P. (2017). 'Using Gaussian Processes to Improve Zero-Shot Learning with Relative Attributes'. In: *Computer Vision – ACCV 2016*. Ed. by S.-H. Lai, V. Lepetit, K. Nishino and Y. Sato. Cham: Springer International Publishing, pp. 150–164. ISBN: 978-3-319-54193-8 (cited on p. 109).

Dong, B., Zhou, P., Yan, S. and Zuo, W. (2022). 'LPT: Long-tailed prompt tuning for image classification'. In: *The Eleventh International Conference on Learning Representations* (cited on p. 31).

Du, Y., Shen, J., Zhen, X. and Snoek, C. G. (2023). 'SuperDisco: Super-Class Discovery Improves Visual Recognition for the Long-Tail'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19944–19954 (cited on p. 31).

Durasov, N., Bagautdinov, T., Baque, P. and Fua, P. (2021). 'Masksembles for Uncertainty Estimation'. In: (cited on p. 31).

Dwivedi, R., Chen, Y., Wainwright, M. J. and Yu, B. (2018). 'Log-concave sampling: Metropolis-Hastings algorithms are fast!' In: *Conference on learning theory*. PMLR, pp. 793–797 (cited on p. 20).

Elyor, K., Tao, X. and Shagong, G. (2017). 'Semantic Autoencoder for Zero-shot Learning'. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cited on p. 5).

Fan, X., Wang, Q., Ke, J., Yang, F., Gong, B. and Zhou, M. (2021). 'Adversarially Adaptive Normalization for Single Domain Generalization'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8208–8217 (cited on p. 3).

Fang, Z., Li, Y., Lu, J., Dong, J., Han, B. and Liu, F. (2022). 'Is out-of-distribution detection learnable?' *Advances in Neural Information Processing Systems*, 35, pp. 37199–37213 (cited on p. 32).

Farhadi, A., Endres, I., Hoiem, D. and Forsyth, D. (2009). 'Describing objects by their attributes'. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1778–1785. DOI: 10.1109/CVPR.2009.5206772 (cited on pp. 108, 114).

Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M. and Mikolov, T. (2013). 'Devise: A deep visual-semantic embedding model'. In: *Advances in neural information processing systems*, pp. 2121–2129 (cited on p. 115).

Gao, T., Han, X., Liu, Z. and Sun, M. (2019). 'Hybrid attention-based prototypical networks for noisy few-shot relation classification'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33, pp. 6407–6414 (cited on pp. 31, 33).

Gardner, J. R., Pleiss, G., Bindel, D., Weinberger, K. Q. and Wilson, A. G. (2018). 'GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration'. *arXiv preprint arXiv:1809.11165* (cited on p. 115).

Garg, S., Balakrishnan, S. and Lipton, Z. (2022). 'Domain adaptation under open set label shift'. *Advances in Neural Information Processing Systems*, 35, pp. 22531–22546 (cited on pp. 3, 4, 13, 27–29, 92).

Garg, S., Erickson, N., Sharpnack, J., Smola, A., Balakrishnan, S. and Lipton, Z. C. (2023). 'Rlsbench: Domain adaptation under relaxed label shift'. In: *International Conference on Machine Learning*. PMLR, pp. 10879–10928 (cited on pp. 27, 28).

Garg, S., Wu, Y., Balakrishnan, S. and Lipton, Z. (2020). 'A unified view of label shift estimation'. *Advances in Neural Information Processing Systems*, 33, pp. 3290–3300 (cited on pp. 3, 12, 26, 29, 38, 42, 52, 55, 57, 63, 67, 100, 123, 124, 148, 161).

Garg, S., Wu, Y., Smola, A. J., Balakrishnan, S. and Lipton, Z. (2021). 'Mixture proportion estimation and PU learning: A modern approach'. *Advances in Neural Information Processing Systems*, 34, pp. 8532–8544 (cited on p. 28).

Geyer, C. J. (1992). 'Practical markov chain monte carlo'. *Statistical science*, pp. 473–483 (cited on p. 19).

Ghosh, K., Bellinger, C., Corizzo, R., Branco, P., Krawczyk, B. and Japkowicz, N. (2024). 'The class imbalance problem in deep learning'. *Machine Learning*, 113(7), pp. 4845–4901 (cited on p. 30).

Goodfellow, I. (2016). *Deep learning* (cited on p. 15).

Goodfellow, I. J., Shlens, J. and Szegedy, C. (2014). 'Explaining and harnessing adversarial examples'. *arXiv preprint arXiv:1412.6572* (cited on p. 32).

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014). *Generative Adversarial Networks* (cited on pp. 26, 107).

Guo, C., Pleiss, G., Sun, Y. and Weinberger, K. Q. (2017). 'On calibration of modern neural networks'. In: *International conference on machine learning*. PMLR, pp. 1321–1330 (cited on pp. 26, 38, 52, 72, 100, 114).

Guo, J., Gong, M., Liu, T., Zhang, K. and Tao, D. (2020). 'Ltf: A label transformation framework for correcting label shift'. In: *International Conference on Machine Learning*. PMLR, pp. 3843–3853 (cited on pp. 26, 27, 29).

Han, P., Ye, C., Zhou, J., Zhang, J., Hong, J. and Li, X. (2024). 'Latent-based Diffusion Model for Long-tailed Recognition'. *arXiv preprint arXiv:2404.04517* (cited on p. 31).

Han, Z., Liang, Z., Yang, F., Liu, L., Li, L., Bian, Y., Zhao, P., Wu, B., Zhang, C. and Yao, J. (2022). 'Umix: Improving importance weighting for subpopulation shift via uncertainty-aware mixup'. *Advances in Neural Information Processing Systems*, 35, pp. 37704–37718 (cited on p. 31).

Han, Z., Fu, Z., Chen, S. and Yang, J. (2021). 'Contrastive Embedding for Generalized Zero-Shot Learning'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2371–2381 (cited on pp. 107, 115, 186).

Han, Z., Fu, Z. and Yang, J. (2020). 'Learning the Redundancy-Free Features for Generalized Zero-Shot Object Recognition'. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (cited on p. 107).

He, H. and Garcia, E. A. (2009). 'Learning from imbalanced data'. *IEEE Transactions on knowledge and data engineering*, 21(9), pp. 1263–1284 (cited on p. 30).

He, H. and Ma, Y. (2013). 'Imbalanced learning: foundations, algorithms, and applications' (cited on p. 30).

He, K., Zhang, X., Ren, S. and Sun, J. (2016). 'Deep residual learning for image recognition'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778 (cited on pp. 15, 47, 70, 91, 148, 161).

Hein, M., Andriushchenko, M. and Bitterwolf, J. (2019). 'Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41–50 (cited on p. 32).

Hendrycks, D., Basart, S., Mazeika, M., Zou, A., Kwon, J., Mostajabi, M., Steinhardt, J. and Song, D. (2019a). 'Scaling out-of-distribution detection for real-world settings'. *arXiv preprint arXiv:1911.11132* (cited on pp. 32, 91, 179).

Hendrycks, D., Basart, S., Mazeika, M., Zou, A., Kwon, J., Mostajabi, M., Steinhardt, J. and Song, D. (2019b). 'Scaling out-of-distribution detection for real-world settings'. *arXiv preprint arXiv:1911.11132* (cited on p. 90).

Hendrycks, D. and Gimpel, K. (2022). 'A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks'. In: *International Conference on Learning Representations* (cited on p. 32).

Hendrycks, D., Mazeika, M., Kadavath, S. and Song, D. (2019). 'Using self-supervised learning can improve model robustness and uncertainty'. *Advances in neural information processing systems*, 32 (cited on p. 32).

Hensman, J., Matthews, A. and Ghahramani, Z. (2015). 'Scalable variational Gaussian process classification' (cited on p. 22).

Ho, J., Jain, A. and Abbeel, P. (2020). 'Denoising diffusion probabilistic models'. *Advances in neural information processing systems*, 33, pp. 6840–6851 (cited on p. 31).

Hoffman, M. D., Gelman, A. et al. (2014). 'The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo.' *J. Mach. Learn. Res.*, 15(1), pp. 1593–1623 (cited on pp. 20, 47, 70).

Hong, J., Fang, P., Li, W., Han, J., Petersson, L. and Harandi, M. (2023). 'Curved geometric networks for visual anomaly recognition'. *IEEE Transactions on Neural Networks and Learning Systems* (cited on p. 32).

Hong, J., Hayder, Z., Han, J., Fang, P., Harandi, M. and Petersson, L. (2023). 'Hyperbolic audio-visual zero-shot learning'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7873–7883 (cited on pp. 33, 119).

Hong, J., Li, W., Han, J., Zheng, J., Fang, P., Harandi, M. and Petersson, L. (2023). 'Goss: Towards generalized open-set semantic segmentation'. *The Visual Computer*, pp. 1–14 (cited on p. 32).

Hong, Y., Han, S., Choi, K., Seo, S., Kim, B. and Chang, B. (2021). 'Disentangling Label Distribution for Long-tailed Visual Recognition'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6626–6636 (cited on pp. 36, 47, 69, 70, 150, 162).

Hsu, Y.-C., Shen, Y., Jin, H. and Kira, Z. (2020). 'Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10951–10960 (cited on pp. 32, 82).

Huang, C., Li, Y., Loy, C. C. and Tang, X. (2016). 'Learning deep representation for imbalanced classification'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5375–5384 (cited on p. 30).

Huang, C., Li, Y., Loy, C. C. and Tang, X. (2019a). 'Deep imbalanced learning for face recognition and attribute prediction'. *IEEE transactions on pattern analysis and machine intelligence*, 42(11), pp. 2781–2794 (cited on p. 1).

Huang, C., Li, Y., Loy, C. C. and Tang, X. (2019b). 'Deep imbalanced learning for face recognition and attribute prediction'. *IEEE transactions on pattern analysis and machine intelligence*, 42(11), pp. 2781–2794 (cited on p. 110).

Huang, H., Wang, C., Yu, P. S. and Wang, C.-D. (2019). 'Generative Dual Adversarial Network for Generalized Zero-Shot Learning'. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cited on pp. 107, 115, 117).

Huang, R. and Li, Y. (2021). 'Mos: Towards scaling out-of-distribution detection for large semantic space'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8710–8719 (cited on p. 90).

Huang, Z., Wang, H., Xing, E. P. and Huang, D. (2020). 'Self-challenging improves cross-domain generalization'. In: *Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, proceedings, part II 16*. Springer, pp. 124–140 (cited on p. 3).

Huynh, D. and Elhamifar, E. (2020). 'Fine-Grained Generalized Zero-Shot Learning via Dense Attribute-Based Attention'. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (cited on pp. 115, 186).

Idelbayev, Y. (n.d.). *Proper ResNet Implementation for CIFAR10/CIFAR100 in PyTorch*. (cited on pp. 47, 70, 148, 161).

Japkowicz, N. and Stephen, S. (2002). 'The class imbalance problem: A systematic study'. *Intelligent data analysis*, 6(5), pp. 429–449 (cited on p. 30).

Johnson, J. M. and Khoshgoftaar, T. M. (2019). 'Survey on deep learning with class imbalance'. *Journal of big data*, 6(1), pp. 1–54 (cited on p. 30).

Joo, T., Chung, U. and Seo, M.-G. (2020). 'Being bayesian about categorical probability'. In: *International Conference on Machine Learning*. PMLR, pp. 4950–4961 (cited on pp. 39, 65).

Kampffmeyer, M., Chen, Y., Liang, X., Wang, H., Zhang, Y. and Xing, E. P. (2019). 'Rethinking Knowledge Graph Propagation for Zero-Shot Learning'. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11487–11496 (cited on pp. 115, 186).

Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J. and Kalantidis, Y. (2020). 'Decoupling Representation and Classifier for Long-Tailed Recognition'. In: *ICLR 2020 : Eighth International Conference on Learning Representations* (cited on p. 31).

Keshari, R., Singh, R. and Vatsa, M. (2020). 'Generalized Zero-Shot Learning via Over-Complete Distribution'. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (cited on pp. 115, 186).

Kim, J., Jeong, J. and Shin, J. (2020). 'M2m: Imbalanced classification via major-to-minor translation'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13896–13905 (cited on p. 31).

Kingma, D. P. and Welling, M. (2013). 'Auto-Encoding Variational Bayes'. *arXiv preprint arXiv:1312.6114* (cited on p. 107).

Kingma, D. P. and Ba, J. (2015). 'Adam: A Method for Stochastic Optimization'. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Y. Bengio and Y. LeCun. (cited on p. 115).

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y. et al. (2023). 'Segment anything'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026 (cited on p. 6).

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y. and Iwasawa, Y. (2022). 'Large language models are zero-shot reasoners'. *Advances in neural information processing systems*, 35, pp. 22199–22213 (cited on p. 33).

Kong, S. and Ramanan, D. (2021). 'Opengan: Open-set recognition via open data generation'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 813–822 (cited on p. 32).

Koßmann, D., Wilhelm, T. and Fink, G. A. (2021). 'Towards tackling multi-label imbalances in remote sensing imagery'. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, pp. 5782–5789 (cited on p. 1).

Krizhevsky, A., Hinton, G. et al. (2009). 'Learning multiple layers of features from tiny images' (cited on pp. 47, 69, 70, 90).

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012). 'Imagenet classification with deep convolutional neural networks'. *Advances in neural information processing systems*, 25 (cited on p. 90).

Kull, M. and Flach, P. (2014). 'Patterns of dataset shift'. In: *First international workshop on learning over multiple contexts (LMCE) at ECML-PKDD*. Vol. 5 (cited on p. 6).

Kutbi, M., Peng, K.-C. and Wu, Z. (2021). 'Zero-shot deep domain adaptation with common representation learning'. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7), pp. 3909–3924 (cited on p. 33).

Lawrence, N. D. (2004). 'Gaussian process models for visualisation of high dimensional data'. *Advances in Neural Information Processing Systems* (cited on p. 22).

Le Cacheux, Y., Le Borgne, H. and Crucianu, M. (2019a). 'From classical to generalized zero-shot learning: A simple adaptation process'. In: *International Conference on Multimedia Modeling*. Springer, pp. 465–477 (cited on pp. 113, 114).

Le Cacheux, Y., Le Borgne, H. and Crucianu, M. (2019b). 'Modeling Inter and Intra-Class Relations in the Triplet Loss for Zero-Shot Learning'. In: *the IEEE International Conference on Computer Vision (ICCV)*. ICCV (cited on pp. 107, 110, 115, 184).

Lee, K., Lee, K., Lee, H. and Shin, J. (2018a). 'A simple unified framework for detecting out-of-distribution samples and adversarial attacks'. *Advances in neural information processing systems*, 31 (cited on p. 32).

Lee, K., Lee, K., Lee, H. and Shin, J. (2018b). 'A simple unified framework for detecting out-of-distribution samples and adversarial attacks'. *Advances in neural information processing systems*, 31 (cited on p. 63).

Lehmann, E. L. and Casella, G. (2006). *Theory of point estimation*. Springer Science & Business Media (cited on p. 16).

Li, J., Jing, M., Lu, K., Ding, Z., Zhu, L. and Huang, Z. (2019). 'Leveraging the invariant side of generative zero-shot learning'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7402–7411 (cited on p. 115).

Li, K., Min, M. R. and Fu, Y. (2019). 'Rethinking Zero-Shot Learning: A Conditional Visual Classification Perspective'. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3583–3592 (cited on pp. 33, 115, 183, 184, 186, 188).

Li, M., Cheung, Y.-m. and Lu, Y. (2022). 'Long-tailed visual recognition via gaussian clouded logit adjustment'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6929–6938 (cited on p. 3).

Li, S., Gong, K., Liu, C. H., Wang, Y., Qiao, F. and Cheng, X. (2021). 'MetaSAug: Meta Semantic Augmentation for Long-Tailed Visual Recognition'. In: *IEEE Conference on Computer Vision and Pattern Recognition* (cited on p. 31).

Liang, S., Li, Y. and Srikant, R. (2017). 'Enhancing the reliability of out-of-distribution image detection in neural networks'. *arXiv preprint arXiv:1706.02690* (cited on pp. 32, 37, 82, 100).

Lipton, Z., Wang, Y.-X. and Smola, A. (2018). 'Detecting and correcting for label shift with black box predictors'. In: *International conference on machine learning*. PMLR, pp. 3122–3130 (cited on pp. 2, 4, 11, 12, 24, 27, 29, 30, 36, 47, 48, 55, 57, 60, 61, 69, 70, 90, 91, 149, 150, 161, 162, 180, 181).

Liu, J., Ye, C., Cui, R. and Barnes, N. (2024). 'Self-Calibrating Vicinal Risk Minimisation for Model Calibration'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3335–3345 (cited on pp. 38, 100).

Liu, J., Ye, C., Wang, S., Cui, R., Zhang, J., Zhang, K. and Barnes, N. (2023). 'Model calibration in dense classification with adaptive label perturbation'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1173–1184 (cited on pp. 38, 100).

Liu, L., Zhou, T., Long, G., Jiang, J., Dong, X. and Zhang, C. (2020). 'Isometric Propagation Network for Generalized Zero-shot Learning'. In: *International Conference on Learning Representations* (cited on pp. 115, 186).

Liu, S., Chen, J., Pan, L., Ngo, C.-W., Chua, T.-S. and Jiang, Y.-G. (2020). 'Hyperbolic Visual Embedding Learning for Zero-Shot Recognition'. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9273–9281 (cited on pp. 116, 187).

Liu, W., Wang, X., Owens, J. and Li, Y. (2020). 'Energy-based out-of-distribution detection'. *Advances in neural information processing systems*, 33, pp. 21464–21475 (cited on p. 32).

Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B. and Stella, X. Y. (2022). 'Open long-tailed recognition in a dynamic world'. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(3), pp. 1836–1851 (cited on p. 30).

Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B. and Yu, S. X. (2019). 'Large-Scale Long-Tailed Recognition in an Open World'. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cited on pp. 3, 5, 30, 31, 35, 47, 55, 108).

Lu, Y., Cheung, Y.-M. and Tang, Y. Y. (2019). 'Bayes imbalance impact index: A measure of class imbalanced data set for classification problem'. *IEEE transactions on neural networks and learning systems*, 31(9), pp. 3525–3539 (cited on p. 3).

Maity, S., Sun, Y. and Banerjee, M. (2022). 'Minimax optimal approaches to the label shift problem in non-parametric settings'. *Journal of Machine Learning Research*, 23(346), pp. 1–45 (cited on pp. 27, 28).

Mangoubi, O. and Smith, A. (2017). 'Rapid mixing of Hamiltonian Monte Carlo on strongly log-concave distributions'. *arXiv preprint arXiv:1708.07114* (cited on p. 46).

Meinke, A., Bitterwolf, J. and Hein, M. (2022). 'Provably Adversarially Robust Detection of Out-of-distribution Data (almost) for free'. *Advances in Neural Information Processing Systems*, 35, pp. 30167–30180 (cited on p. 92).

Meinke, A. and Hein, M. (2019). 'Towards neural networks that provably know when they don't know'. In: *International Conference on Learning Representations* (cited on p. 32).

Meng, Y. and Guo, Y. (2017). 'Zero-Shot Classification with Discriminative Semantic Representation Learning'. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cited on p. 33).

Menon, A. K., Jayasumana, S., Rawat, A. S., Jain, H., Veit, A. and Kumar, S. (2020). 'Long-tail learning via logit adjustment'. In: *International Conference on Learning Representations* (cited on p. 31).

Miller, J. P., Taori, R., Raghunathan, A., Sagawa, S., Koh, P. W., Shankar, V., Liang, P., Carmon, Y. and Schmidt, L. (2021). 'Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization'. In: *International Conference on Machine Learning*. PMLR, pp. 7721–7735 (cited on p. 32).

Min, S., Yao, H., Xie, H., Wang, C., Zha, Z. and Zhang, Y. (2020). 'Domain-Aware Visual Bias Eliminating for Generalized Zero-Shot Learning'. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12661–12670 (cited on pp. 107, 115, 117, 183, 184, 186).

Moon, T. K. (1996). 'The expectation-maximization algorithm'. *IEEE Signal processing magazine*, 13(6), pp. 47–60 (cited on p. 21).

Morteza, P. and Li, Y. (2022). 'Provable guarantees for understanding out-of-distribution detection'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 7, pp. 7831–7840 (cited on p. 63).

Mukherjee, T. and Hospedales, T. (2016). 'Gaussian Visual-Linguistic Embedding for Zero-Shot Recognition'. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 912–918. DOI: 10.18653/v1/D16-1089 (cited on p. 111).

Müller, R., Kornblith, S. and Hinton, G. E. (2019). 'When does label smoothing help?' *Advances in neural information processing systems*, 32 (cited on p. 31).

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press (cited on pp. 17, 19, 87, 169).

Narayan, S., Gupta, A., Khan, F. S., Snoek, C. G. and Shao, L. (2020). 'Latent Embedding Feedback and Discriminative Features for Zero-Shot Classification'. In: *ECCV* (cited on pp. 115, 186).

Neal, R. (2011). 'MCMC Using Hamiltonian Dynamics'. *Handbook of Markov Chain Monte Carlo*, pp. 113–162 (cited on pp. 20, 45).

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A. Y. et al. (2011). 'Reading digits in natural images with unsupervised feature learning'. In: *NIPS workshop on deep learning and unsupervised feature learning*. Vol. 2011. 2. Granada, p. 4 (cited on p. 90).

Norouzi, M., Mikolov, T., Bengio, S., Singer, Y., Shlens, J., Frome, A., Corrado, G. S. and Dean, J. (2013). 'Zero-Shot Learning by Convex Combination of Semantic Embeddings'. *arXiv preprint arXiv:1312.5650* (cited on pp. 33, 107).

Pan, S. J. and Yang, Q. (2009). 'A survey on transfer learning'. *IEEE Transactions on knowledge and data engineering*, 22(10), pp. 1345–1359 (cited on p. 3).

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L. and Lerer, A. (2017). 'Automatic differentiation in PyTorch' (cited on pp. 47, 70, 91, 115, 148, 161).

Patterson, G. and Hays, J. (2012). 'Sun attribute database: Discovering, annotating, and recognizing scene attributes'. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 2751–2758 (cited on p. 114).

Peng, H., Sun, M. and Li, P. (2022). 'Optimal Transport for Long-Tailed Recognition with Learnable Cost Matrix'. In: *International Conference on Learning Representations*. (cited on p. 31).

Peng, K.-C., Wu, Z. and Ernst, J. (2018). 'Zero-shot deep domain adaptation'. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 764–781 (cited on p. 33).

Podkopaev, A. and Ramdas, A. (2021). 'Distribution-free uncertainty quantification for classification under label shift'. In: *Uncertainty in Artificial Intelligence*. PMLR, pp. 844–853 (cited on pp. 27, 72).

Pourpanah, F., Abdar, M., Luo, Y., Zhou, X., Wang, R., Lim, C. P., Wang, X.-Z. and Wu, Q. J. (2022). 'A review of generalized zero-shot learning methods'. *IEEE transactions on pattern analysis and machine intelligence*, 45(4), pp. 4051–4070 (cited on p. 33).

Qian, Y.-Y., Bai, Y., Zhang, Z.-Y., Zhao, P. and Zhou, Z.-H. (2023). 'Handling New Class in Online Label Shift'. In: *2023 IEEE International Conference on Data Mining (ICDM)*. IEEE, pp. 1283–1288 (cited on pp. 27, 28).

Qiao, F., Zhao, L. and Peng, X. (2020). 'Learning to learn single domain generalization'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12556–12565 (cited on p. 3).

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J. et al. (2021). 'Learning transferable visual models from natural language supervision'. In: *International conference on machine learning*. PMLR, pp. 8748–8763 (cited on pp. 33, 119).

Rasmussen, C., Williams, C., Press, M., Bach, F. and (Firm), P. (2006). *Gaussian Processes for Machine Learning*. Adaptive computation and machine learning. MIT Press. ISBN: 9780262182539. (cited on p. 108).

Rezvani, S. and Wang, X. (2023). 'A broad review on class imbalance learning techniques'. *Applied Soft Computing*, 143, p. 110415 (cited on p. 30).

Robbins, H. E. (1992). 'An empirical Bayes approach to statistics'. In: *Breakthroughs in statistics*. Springer, pp. 388–394 (cited on p. 42).

Russakovsky, O. et al. (2015a). 'ImageNet Large Scale Visual Recognition Challenge'. *International Journal of Computer Vision (IJCV)*, 115(3), pp. 211–252. DOI: 10.1007/s11263-015-0816-y (cited on pp. 15, 90).

Russakovsky, O. et al. (2015b). 'ImageNet Large Scale Visual Recognition Challenge'. *International Journal of Computer Vision (IJCV)*, 115(3), pp. 211–252. DOI: 10.1007/s11263-015-0816-y (cited on pp. 47, 69, 70).

Saerens, M., Latinne, P. and Decaestecker, C. (2002). 'Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure'. *Neural computation*, 14(1), pp. 21–41 (cited on pp. 4, 25–27, 29, 48, 63, 67, 70, 74, 86, 87, 91, 122–124, 144, 145, 169, 180).

Salehi, M., Mirzaei, H., Hendrycks, D., Li, Y., Rohban, M. H. and Sabokrou, M. (2021). 'A unified survey on anomaly, novelty, open-set, and out-of-distribution detection: Solutions and future challenges'. *arXiv preprint arXiv:2110.14051* (cited on p. 32).

Santurkar, S., Tsipras, D. and Madry, A. (2020). 'Breeds: Benchmarks for subpopulation shift'. *arXiv preprint arXiv:2008.04859* (cited on p. 6).

Sastry, C. S. and Oore, S. (2020). 'Detecting out-of-distribution examples with gram matrices'. In: *International Conference on Machine Learning*. PMLR, pp. 8491–8501 (cited on p. 32).

Schönfeld, E., Ebrahimi, S., Sinha, S., Darrell, T. and Akata, Z. (2019). 'Generalized Zero- and Few-Shot Learning via Aligned Variational Autoencoders'. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8239–8247 (cited on pp. 115, 117, 118).

Shapiro, A., Dentcheva, D. and Ruszczynski, A. (2021). *Lectures on stochastic programming: modeling and theory*. SIAM (cited on p. 28).

Shen, Y., Qin, J., Huang, L., Liu, L., Zhu, F. and Shao, L. (2020). 'Invertible Zero-Shot Recognition Flows'. In: *European Conference on Computer Vision*, pp. 614–631 (cited on pp. 115, 186).

Shimodaira, H. (2000). 'Improving predictive inference under covariate shift by weighting the log-likelihood function'. *Journal of statistical planning and inference*, 90(2), pp. 227–244 (cited on p. 3).

Simonyan, K. and Zisserman, A. (2015). 'Very Deep Convolutional Networks for Large-Scale Image Recognition'. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Y. Bengio and Y. LeCun. (cited on p. 15).

Skorokhodov, I. and Elhoseiny, M. (2021). 'Class normalization for (continual)? generalized zero-shot learning'. In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cited on pp. 113, 115–118).

Snell, J., Swersky, K. and Zemel, R. (2017a). 'Prototypical networks for few-shot learning'. *Advances in neural information processing systems*, 30 (cited on pp. 55, 63).

Snell, J., Swersky, K. and Zemel, R. S. (2017b). 'Prototypical Networks for Few-shot Learning'. In: *NIPS* (cited on p. 33).

Sohn, K. (2016). 'Improved Deep Metric Learning with Multi-class N-pair Loss Objective'. In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon and R. Garnett. Vol. 29. Curran Associates, Inc. (cited on p. 110).

Song, J., Meng, C. and Ermon, S. (2020). 'Denoising diffusion implicit models'. *arXiv preprint arXiv:2010.02502* (cited on p. 31).

Storkey, A. (2009). 'When training and test sets are different: characterizing learning transfer'. *Dataset shift in machine learning*, 30, pp. 3–28 (cited on p. 4).

Sugiyama, M., Krauledat, M. and Müller, K.-R. (2007). 'Covariate shift adaptation by importance weighted cross validation.' *Journal of Machine Learning Research*, 8(5) (cited on p. 6).

Sun, Y., Guo, C. and Li, Y. (2021). 'React: Out-of-distribution detection with rectified activations'. *Advances in Neural Information Processing Systems*, 34, pp. 144–157 (cited on pp. 32, 91, 179).

Sun, Y., Ming, Y., Zhu, X. and Li, Y. (2022). 'Out-of-distribution detection with deep nearest neighbors'. In: *International Conference on Machine Learning*. PMLR, pp. 20827–20840 (cited on pp. 32, 91, 179).

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H. S. and Hospedales, T. M. (2018). 'Learning to Compare: Relation Network for Few-Shot Learning'. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208 (cited on pp. 115, 186).

Suresh, T., Brijet, Z. and Subha, T. (2023). 'Imbalanced medical disease dataset classification using enhanced generative adversarial network'. *Computer Methods in Biomechanics and Biomedical Engineering*, 26(14), pp. 1702–1718 (cited on p. 1).

Suzuki, T. (2012). 'PAC-Bayesian Bound for Gaussian Process Regression and Multiple Kernel Additive Model'. *J. Mach. Learn. Res. Wrkshp Conf. Proc.*, 23 (cited on p. 108).

Tachet des Combes, R., Zhao, H., Wang, Y.-X. and Gordon, G. J. (2020). 'Domain adaptation with conditional distribution matching and generalized label shift'. *Advances in Neural Information Processing Systems*, 33, pp. 19276–19289 (cited on pp. 25, 27, 29, 48, 70).

Tack, J., Mo, S., Jeong, J. and Shin, J. (2020). 'Csi: Novelty detection via contrastive learning on distributionally shifted instances'. *Advances in neural information processing systems*, 33, pp. 11839–11852 (cited on p. 32).

Thrun, S. and Pratt, L. (1998). 'Learning to learn: Introduction and overview'. In: *Learning to learn*. Springer, pp. 3–17 (cited on p. 3).

Thulasidasan, S., Chennupati, G., Bilmes, J. A., Bhattacharya, T. and Michalak, S. (2019). 'On Mixup Training: Improved Calibration and Predictive Uncertainty for Deep Neural Networks'. *Advances in Neural Information Processing Systems*, 32 (cited on pp. 148, 161).

Tian, J., Liu, Y.-C., Glaser, N., Hsu, Y.-C. and Kira, Z. (2020). 'Posterior re-calibration for imbalanced datasets'. *Advances in Neural Information Processing Systems*, 33, pp. 8101–8113 (cited on p. 27).

Tian, Q., Zhang, X. and Zhao, J. (2023). 'ELSA: Efficient label shift adaptation through the lens of semiparametric models'. In: *International Conference on Machine Learning*. PMLR, pp. 34120–34142 (cited on pp. 26, 27, 29).

Törnberg, P. (2023). 'Chatgpt-4 outperforms experts and crowd workers in annotating political twitter messages with zero-shot learning'. *arXiv preprint arXiv:2304.06588* (cited on p. 33).

Tsoumakas, G. and Katakis, I. (2008). 'Multi-label classification: An overview'. *Data Warehousing and Mining: Concepts, Methodologies, Tools, and Applications*, pp. 64–74 (cited on p. 6).

Tu, S. (2014). 'The dirichlet-multinomial and dirichlet-categorical models for bayesian inference'. *Computer Science Division, UC Berkeley*, 2 (cited on pp. 39, 65).

Upton, G. and Cook, I. (2014). *A dictionary of statistics 3e*. Oxford University Press, USA (cited on p. 16).

Urtasun, R. and Darrell, T. (2007). 'Discriminative Gaussian process latent variable model for classification'. In: *Proceedings of the 24th international conference on Machine learning*, pp. 927–934 (cited on p. 22).

Vapnik, V. (1991). 'Principles of risk minimization for learning theory'. *Advances in neural information processing systems*, 4 (cited on p. 15).

Varadhan, R. and Roland, C. (2008). 'Simple and globally convergent methods for accelerating the convergence of any EM algorithm'. *Scandinavian Journal of Statistics*, 35(2), pp. 335–353 (cited on p. 123).

Vaze, S., Han, K., Vedaldi, A. and Zisserman, A. (2021). 'Open-Set Recognition: A Good Closed-Set Classifier is All You Need'. In: *International Conference on Learning Representations* (cited on p. 32).

Verma, V. K. and Rai, P. (2017). 'A Simple Exponential Family Framework for Zero-Shot Learning'. In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by M. Ceci, J. Hollmén, L. Todorovski, C. Vens and S. Džeroski. Cham: Springer International Publishing, pp. 792–808. ISBN: 978-3-319-71246-8 (cited on pp. 33, 107, 109, 115).

Vershynin, R. (2018). *High-dimensional probability: An introduction with applications in data science*. Vol. 47. Cambridge university press (cited on pp. 84, 164).

Vucetic, S. and Obradovic, Z. (2001). 'Classification on data with biased class distribution'. In: *Machine Learning: ECML 2001: 12th European Conference on Machine Learning Freiburg, Germany, September 5–7, 2001 Proceedings 12*. Springer, pp. 527–538 (cited on p. 4).

Wah, C., Branson, S., Welinder, P., Perona, P. and Belongie, S. (2011). *The Caltech-UCSD Birds-200-2011 Dataset*. Tech. rep. CNS-TR-2011-001. California Institute of Technology (cited on p. 114).

Wang, H., Li, Z., Feng, L. and Zhang, W. (2022a). 'Vim: Out-of-distribution with virtual-logit matching'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4921–4930 (cited on p. 32).

Wang, H., Li, Z., Feng, L. and Zhang, W. (2022b). 'Vim: Out-of-distribution with virtual-logit matching'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4921–4930 (cited on p. 90).

Wang, S., Liu, W., Wu, J., Cao, L., Meng, Q. and Kennedy, P. J. (2016). 'Training deep neural networks on imbalanced data sets'. In: *2016 international joint conference on neural networks (IJCNN)*. IEEE, pp. 4368–4374 (cited on p. 3).

Wang, W., Zheng, V. W., Yu, H. and Miao, C. (2019). 'A survey of zero-shot learning: Settings, methods, and applications'. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2), pp. 1–37 (cited on p. 33).

Wang, Y.-X., Ramanan, D. and Hebert, M. (2017). 'Learning to model the tail'. *Advances in Neural Information Processing Systems*, 30 (cited on p. 30).

Wang, X., Lian, L., Miao, Z., Liu, Z. and Yu, S. (2020). 'Long-tailed Recognition by Routing Diverse Distribution-Aware Experts'. In: *International Conference on Learning Representations* (cited on pp. 31, 47, 69).

Wang, Y., Pan, X., Song, S., Zhang, H., Huang, G. and Wu, C. (2019). 'Implicit semantic data augmentation for deep networks'. *Advances in Neural Information Processing Systems*, 32 (cited on p. 31).

Wang, Z., Luo, Y., Qiu, R., Huang, Z. and Baktashmotlagh, M. (2021). 'Learning to diversify for single domain generalization'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 834–843 (cited on p. 3).

Wen, Y., Zhang, K., Li, Z. and Qiao, Y. (2016). 'A discriminative feature learning approach for deep face recognition'. In: *European conference on computer vision*. Springer, pp. 499–515 (cited on p. 111).

Wen, Y., Tran, D. and Ba, J. (2019). 'BatchEnsemble: an Alternative Approach to Efficient Ensemble and Lifelong Learning'. In: *International Conference on Learning Representations* (cited on p. 31).

Williams, C. K. and Rasmussen, C. E. (2006). *Gaussian processes for machine learning*. Vol. 2. 3. MIT press Cambridge, MA (cited on p. 22).

Wu, J., Zhang, T., Zha, Z.-J., Luo, J., Zhang, Y. and Wu, F. (2020). 'Self-Supervised Domain-Aware Generative Network for Generalized Zero-Shot Learning'. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12767–12776 (cited on pp. 116, 187).

Wu, R., Guo, C., Su, Y. and Weinberger, K. Q. (2021). 'Online adaptation to label distribution shift'. *Advances in Neural Information Processing Systems*, 34, pp. 11340–11351 (cited on pp. 4, 27, 28, 124).

Wu, Z., Guo, K., Luo, E., Wang, T., Wang, S., Yang, Y., Zhu, X. and Ding, R. (2024). 'Medical long-tailed learning for imbalanced data: bibliometric analysis'. *Computer Methods and Programs in Biomedicine*, p. 108106 (cited on p. 30).

Xian, Y., Lampert, C. H., Schiele, B. and Akata, Z. (2019). 'Zero-Shot Learning—A Comprehensive Evaluation of the Good, the Bad and the Ugly'. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9), pp. 2251–2265. DOI: 10.1109/TPAMI.2018.2857768 (cited on pp. 5, 33, 108, 114, 115, 183, 184).

Xian, Y., Lorenz, T., Schiele, B. and Akata, Z. (2018). 'Feature Generating Networks for Zero-Shot Learning'. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5542–5551 (cited on p. 107).

Xian, Y., Sharma, S., Schiele, B. and Akata, Z. (2019). 'F-VAEGAN-D2: A Feature Generating Framework for Any-Shot Learning'. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10267–10276 (cited on pp. 115, 186).

Xiang, L., Ding, G. and Han, J. (2020). 'Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification'. In: *European Conference on Computer Vision*. Springer, pp. 247–263 (cited on p. 31).

Xu, Z., Chai, Z. and Yuan, C. (2021). 'Towards Calibrated Model for Long-Tailed Visual Recognition from Prior Perspective'. *Advances in Neural Information Processing Systems*, 34 (cited on pp. 5, 30, 31, 37, 100).

Yang, J., Wang, P., Zou, D., Zhou, Z., Ding, K., Peng, W., Wang, H., Chen, G., Li, B., Sun, Y. et al. (2022). 'Openood: Benchmarking generalized out-of-distribution detection'. *Advances in Neural Information Processing Systems*, 35, pp. 32598–32611 (cited on pp. 179, 180).

Yang, J., Zhou, K., Li, Y. and Liu, Z. (2024). 'Generalized out-of-distribution detection: A survey'. *International Journal of Computer Vision*, pp. 1–28 (cited on p. 32).

Yang, L., Jiang, H., Song, Q. and Guo, J. (2022). 'A survey on long-tailed visual recognition'. *International Journal of Computer Vision*, 130(7), pp. 1837–1872 (cited on p. 30).

Yao, Y., Liu, T., Han, B., Gong, M., Niu, G., Sugiyama, M. and Tao, D. (2020). 'Rethinking class-prior estimation for positive-unlabeled learning'. *arXiv preprint arXiv:2002.03673* (cited on p. 28).

Ye, C., Tsuchida, R., Petersson, L. and Barnes, N. (2024). 'Label Shift Estimation for Class-Imbalance Problem: A Bayesian Approach'. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1073–1082 (cited on pp. 30, 37, 89–92, 100, 101, 170, 181).

Yoon, K. and Kwek, S. (2005). 'An unsupervised learning approach to resolving the data imbalanced issue in supervised learning problems in functional genomics'. In: *Fifth International Conference on Hybrid Intelligent Systems (HIS'05)*. IEEE, 6–pp (cited on p. 1).

Yu, Y., Ji, Z., Han, J. and Zhang, Z. (2020). 'Episode-Based Prototype Generating Network for Zero-Shot Learning'. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (cited on pp. 115, 117, 183, 184, 186).

Yue, X., Mou, N., Wang, Q. and Zhao, L. (2024). 'Revisiting Adversarial Training under Long-Tailed Distributions'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24492–24501 (cited on p. 31).

Yue, Z., Wang, T., Zhang, H., Sun, Q. and Hua, X. (2021). 'Counterfactual Zero-Shot and Open-Set Visual Recognition'. *CoRR*, abs/2103.00887. (cited on pp. 115–117).

Zhang, H., Cisse, M., Dauphin, Y. N. and Lopez-Paz, D. (2018). 'mixup: Beyond Empirical Risk Minimization'. In: *International Conference on Learning Representations*. (cited on pp. 15, 29, 31, 148, 161).

Zhang, J., Fan, D.-P., Dai, Y., Anwar, S., Saleh, F., Aliakbarian, S. and Barnes, N. (2021). 'Uncertainty inspired RGB-D saliency detection'. *IEEE transactions on pattern analysis and machine intelligence*, 44(9), pp. 5761–5779 (cited on p. 6).

Zhang, J., Yang, J. et al. (2023). 'OpenOOD v1.5: Enhanced Benchmark for Out-of-Distribution Detection'. *arXiv preprint arXiv:2306.09301* (cited on pp. 87, 90, 91, 179).

Zhang, J., Menon, A. K., Veit, A., Bhojanapalli, S., Kumar, S. and Sra, S. (2021). 'COPING WITH LABEL SHIFT VIA DISTRIBUTIONALLY ROBUST OPTIMISATION'. In: *International Conference on Learning Representations (ICLR)* (cited on pp. 27, 28).

Zhang, K., Schölkopf, B., Muandet, K. and Wang, Z. (2013). 'Domain adaptation under target and conditional shift'. In: *International conference on machine learning*. PMLR, pp. 819–827 (cited on p. 4).

Zhang, Y., Kang, B., Hooi, B., Yan, S. and Feng, J. (2023). 'Deep long-tailed learning: A survey'. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9), pp. 10795–10816 (cited on p. 30).

Zhao, E., Liu, A., Anandkumar, A. and Yue, Y. (2021). 'Active learning under label shift'. In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 3412–3420 (cited on pp. 27, 28, 124).

Zheng, Q., Hong, J. and Farazi, M. (2023). 'A generative approach to audio-visual generalized zero-shot learning: Combining contrastive and discriminative techniques'. In: *2023 International Joint Conference on Neural Networks (IJCNN)*. IEEE, pp. 1–8 (cited on pp. 33, 119).

Zhong, Z., Cui, J., Liu, S. and Jia, J. (2021). 'Improving calibration for long-tailed recognition'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16489–16498 (cited on pp. 31, 47, 69).

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A. and Torralba, A. (2017). 'Places: A 10 million Image Database for Scene Recognition'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (cited on pp. 47, 69, 70, 90).

Zhou, B., Cui, Q., Wei, X.-S. and Chen, Z.-M. (2020). '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*, pp. 9719–9728 (cited on p. 31).

Zhu, L. and Yang, Y. (2020). 'Inflated episodic memory with region self-attention for long-tailed visual recognition'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4344–4353 (cited on p. 31).

Appendix A

Appendix for Chapter 3

A.1 Mathematical Proofs

A.1.1 Proof of Proposition 1 Convexity (See page 41)

Proof. We assume the parameter $\boldsymbol{\pi}$ of target label distribution $p_t(y = \cdot) = \boldsymbol{\pi}$ follows a Dirichlet prior distribution $\boldsymbol{\pi} \sim \text{Dir}(K, \boldsymbol{\alpha})$, where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, ...\alpha_K]^T$ is the parameter of the prior. For unlabeled sample $\mathcal{D}^t = \{x_i^t\}_{i=1}^{N^t}$ drawn i.i.d. from target distribution $p_t(x, y)$, we want to find parameter $\boldsymbol{\pi} \in \Delta^{K-1}$ that maximizes the posterior $p(\boldsymbol{\pi}|\mathcal{D}^t, \boldsymbol{\alpha})$, or equivalently minimizes the negative log posterior:

$$\begin{aligned} \boldsymbol{\pi}^* &= \arg\min_{\boldsymbol{\pi} \in \Delta^{K-1}} -\log p(\boldsymbol{\pi}|\mathcal{D}^t, \boldsymbol{\alpha}) \\ &= \arg\min_{\boldsymbol{\pi} \in \Delta^{K-1}} -\left(\log \prod_{i=1}^{N^t} p_t(x_i^t|\boldsymbol{\pi}) + \log p(\boldsymbol{\pi}|\boldsymbol{\alpha}) + Const \right) \end{aligned} \tag{A.1}$$

where $Const$ includes all terms that can be treated as a constant w.r.t. $\boldsymbol{\pi}$. $p(\boldsymbol{\pi}|\boldsymbol{\alpha})$ is the Dirichlet prior, with $\log p(\boldsymbol{\pi}|\boldsymbol{\alpha})$ as a concave function of $\boldsymbol{\pi}$:

$$\log p(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \log \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^{K} \pi_i^{\alpha_i - 1} = \sum_{i=1}^{K} (\alpha_i - 1) \log \pi_i + Const \tag{A.2}$$

with $\mathrm{B}(\boldsymbol{\alpha})$ as the normalization constant for a given $\boldsymbol{\alpha}$. Here we require $\alpha_i - 1 > 0, i = 1, 2...K$.

Suppose the equality in Eq. (A.3) below holds for the source label distribution $p_s(y = j)$, classifier f and target label distribution $p_t(y = j|\boldsymbol{\pi})$:

$$\begin{aligned} p_s(y = j) &= c_j > 0 \\ p_s(y = j|x) &= f(x)_j \\ p_t(y = j|\boldsymbol{\pi}) &= \pi_j \end{aligned} \tag{A.3}$$

Alexandari et al. (2020) has proved that under label shift, if Eq. (A.3) holds for $(x,i) \in \mathcal{X} \times \mathcal{Y}$, then the negative log likelihood defined in Eq. (A.4) is convex.

$$-\log L(\boldsymbol{\pi}; \mathcal{D}^t) = -\log p(\mathcal{D}^t|\boldsymbol{\pi}) = -\log \prod_{i=1}^{N^t} p_t(x_i^t|\boldsymbol{\pi}) \tag{A.4}$$

Thus the MAP objective in Eq. (A.1) can be rewritten as:

$$\boldsymbol{\pi}^* = \underset{\boldsymbol{\pi} \in \Delta^{K-1}}{\arg\min} - \left(\log L(\boldsymbol{\pi}; \mathcal{D}^t) + \sum_{i=1}^{K} (\alpha_i - 1) \log \pi_i + Const \right). \tag{A.5}$$

The objective function in Eq. (A.5) is adding Eq. (A.4) with extra term $\sum_{l=1}^{K}(\alpha_l - 1)\log \pi_l$, which is a strictly convex function. Given the constraints $\boldsymbol{\pi} \in \Delta^{K-1} \subseteq \mathbb{R}^K$ defines a convex set, the above optimization problem is a convex optimization problem with a unique global minima. □

A.1.2 Proof of Proposition 1 EM algorithm (See page 41)

Proof. We follow the standard EM derivation procedure. We first derive the form of complete posterior $p(\boldsymbol{\pi}|\mathcal{D}^t, \mathbb{Y}, \boldsymbol{\alpha})$ with unobserved latent variable $\mathbb{Y}$ based on the original posterior $p(\boldsymbol{\pi}|\mathcal{D}^t, \boldsymbol{\alpha})$. Based on same assumptions made by MLLS Alexandari et al. (2020); Saerens et al. (2002), we construct analytical expression of $Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(m)})$ function for E-Step in EM. Finally, we optimize $Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(m)})$ w.r.t. $\boldsymbol{\pi}$ to find the M-Step.

The EM algorithm optimizes the original posterior $p(\boldsymbol{\pi}|\mathcal{D}^t, \boldsymbol{\alpha})$ with an iterative procedure. Target labels $\mathbb{Y} = \{y_i^t | \{y_i^t, x_i^t\} \sim p_t(x,y) x_i^t \in \mathcal{D}^t\}$ of input image $\mathcal{D}^t$ are treated as the unobserved latent variables. The complete posterior $p(\boldsymbol{\pi}|\mathcal{D}^t, \mathbb{Y}, \boldsymbol{\alpha})$ that includes latent variable $\mathbb{Y}$ can be written as:

$$\begin{aligned}
p(\boldsymbol{\pi}|\mathcal{D}^t, \mathbb{Y}, \boldsymbol{\alpha}) &= \frac{p(\boldsymbol{\pi}|\boldsymbol{\alpha}) p(\mathcal{D}^t, \mathbb{Y}|\boldsymbol{\pi})}{\int_{\boldsymbol{\pi}} p(\boldsymbol{\pi}|\boldsymbol{\alpha}) p(\mathcal{D}^t, \mathbb{Y}|\boldsymbol{\pi}) d\boldsymbol{\pi}} \\
&= \frac{1}{Z} p(\boldsymbol{\pi}|\boldsymbol{\alpha}) \prod_{i=1}^{N^t} p_t(x_i^t, y_i^t|\boldsymbol{\pi}) \\
&= \frac{1}{Z} p(\boldsymbol{\pi}|\boldsymbol{\alpha}) \prod_{i=1}^{N^t} p_t(x_i^t|y_i^t) p_t(y = y_i^t|\boldsymbol{\pi}) \\
&= \frac{1}{Z} p(\boldsymbol{\pi}|\boldsymbol{\alpha}) \prod_{i=1}^{N^t} p_s(x_i^t|y_i^t) p_t(y = y_i^t|\boldsymbol{\pi}) \\
&= \frac{1}{Z} p(\boldsymbol{\pi}|\boldsymbol{\alpha}) \prod_{i=1}^{N^t} \frac{p_t(y = y_i^t|\boldsymbol{\pi})}{p_s(y = y_i^t)} p_s(y = y_i^t|x_i^t) p_s(x_i^t)
\end{aligned} \tag{A.6}$$

where Z is the integral in the denominator that has integrated out $\boldsymbol{\pi}$ and can be treated as a constant w.r.t. $\boldsymbol{\pi}$. Further, $p(\boldsymbol{\pi}|\boldsymbol{\alpha})$ is the Dirichlet prior of $\boldsymbol{\pi}$ that has the

expression:

$$p(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^{K} \pi_i^{\alpha_i - 1} \tag{A.7}$$

with $\mathrm{B}(\boldsymbol{\alpha})$ as the normalization constant for a given $\boldsymbol{\alpha}$ and $\alpha_i - 1 > 0, i = 1, 2...K$.

In the **E-Step**, given the parameter $\boldsymbol{\pi}^{(m)}$ in the m^{th} iteration, with the complete posterior $p(\boldsymbol{\pi}|\mathcal{D}^t, \mathbb{Y}, \boldsymbol{\alpha})$ defined in Eq. (A.6), the $Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(m)})$ can be written as:

$$\begin{aligned}
Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(m)}) &= \mathop{\mathbb{E}}_{\mathbb{Y}|\mathcal{D}^t, \boldsymbol{\pi}^{(m)}} \left[\log p(\boldsymbol{\pi}|\mathcal{D}^t, \mathbb{Y}, \boldsymbol{\alpha})\right] \\
&= \mathop{\mathbb{E}}_{\mathbb{Y}|\mathcal{D}^t, \boldsymbol{\pi}^{(m)}} \left[\log \left(p(\boldsymbol{\pi}|\boldsymbol{\alpha}) \prod_{i=1}^{N^t} \frac{p_t(y = y_i^t|\boldsymbol{\pi})}{p_s(y = y_i^t)} p_s(y = y_i^t|x_i^t) p_s(x_i^t) \right) \right. \\
&\quad \left. + Const \right] \\
&= \mathop{\mathbb{E}}_{\mathbb{Y}|\mathcal{D}^t, \boldsymbol{\pi}^{(m)}} \left[\sum_{i=1}^{N^t} \log \prod_{j=1}^{K} \pi_j^{\mathbb{I}_j(y_i^t)} \right] + \sum_{l=1}^{K} (\alpha_l - 1) \log \pi_l + Const \\
&= \sum_{i=1}^{N^t} \sum_{j=1}^{K} \mathop{\mathbb{E}}_{\mathbb{Y}|\mathcal{D}^t, \boldsymbol{\pi}^{(m)}} \left[\mathbb{I}_j(y_i^t)\right] \log \pi_j + \sum_{l=1}^{K} (\alpha_l - 1) \log \pi_l + Const \\
&= \sum_{j=1}^{K} \sum_{i=1}^{N^t} p_t(y = j|x_i^t, \boldsymbol{\pi}^{(m)}) \log \pi_j + \sum_{l=1}^{K} (\alpha_l - 1) \log \pi_l + Const
\end{aligned} \tag{A.8}$$

where $\mathbb{I}_j(y_i^t)$ is the indicator function. In the above derivation, all terms that are irrelevant to $\boldsymbol{\pi}$ are moved in the term $Const$. For example, $\mathbb{E}_{\mathbb{Y}|\mathcal{D}^t, \boldsymbol{\pi}^{(m)}}[p_s(y = y_i^t)]$ and $\log \mathrm{B}(\boldsymbol{\alpha})$.

Note that in label shift, Saerens et al. (2002) proved that under label shift assumption (Assumption 1), $p_t(y = j|x_i^t, \boldsymbol{\pi})$ can be written as:

$$p_t(y = j|x_i^t, \boldsymbol{\pi}^{(m)}) = \frac{\frac{p_t(y=j;\boldsymbol{\pi}^{(m)})}{p_s(y=j)} p_s(y = j|x_i^t)}{\sum_{l=1}^{K} \frac{p_t(y=l;\boldsymbol{\pi}^{(m)})}{p_s(y=l)} p_s(y = l|x_i^t)} \tag{A.9}$$

We can substitute analytical expression of each probability into the equation with Eq. (A.3):

$$g_{ij}^{(m)} := p_t(y = j|x_i^t, \boldsymbol{\pi}^{(m)}) = \frac{\frac{\pi_j}{c_j} f(x_i^t)_j}{\sum_{l=1}^{K} \frac{\pi_l^{(m)}}{c_l} f(x_i^t)_l} \tag{A.10}$$

Then the $Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(m)})$ can be rewritten as:

$$Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(m)}) = \sum_{j=1}^{K} \sum_{i=1}^{N^t} g_{ij}^{(m)} \log \pi_j + \sum_{l=1}^{K} (\alpha_l - 1) \log \pi_l + Const \tag{A.11}$$

In the **M-step**, we solve the optimization objective with respect to $\boldsymbol{\pi}$:

$$\boldsymbol{\pi}^{(t+1)} = \arg\max_{\boldsymbol{\pi} \in \Delta^{K-1}} Q(\boldsymbol{\pi} | \boldsymbol{\pi}^{(m)}) \tag{A.12}$$

By substitution, the objective can be rewritten as:

$$\begin{cases} \min\limits_{\boldsymbol{\pi}} -\sum\limits_{i=1}^{N^t}\sum\limits_{j=1}^{K} g_{ij}^{(m)} \log \pi_j - \sum\limits_{l=1}^{K} (\alpha_l - 1) \log \pi_l \\ \text{s.t: } \sum\limits_{j=1}^{K} \pi_j = 1 \text{ and } \pi_i \geq 0, i \in [1, 2, ...K] \end{cases} \tag{A.13}$$

Convexity The objective we want to optimize is just a linear combination of $\log \pi_i$, which is a concave function w.r.t. $\boldsymbol{\pi}$. Knowing that the constraints define a convex set on $\mathbb{R}^K$, the above optimization problem is also a convex optimization problem and every local minima is a global minima.

Optimization without inequality constraints With only equality constraints, standard Lagrangian Multiplier method can be applied. The Lagrangian can be written as:

$$\mathcal{L}(\boldsymbol{\pi}, \lambda) = \sum_{i=1}^{N^t}\sum_{j=1}^{K} g_{ij}^{(m)} \log \pi_j + \sum_{j=1}^{K} (\alpha_j - 1) \log \pi_j + \lambda (1 - \sum_{j=1}^{K} \pi_j) \tag{A.14}$$

The optimal $\boldsymbol{\pi}$ can be found by taking all the partial derivative of $\mathcal{L}(\boldsymbol{\pi}, \lambda)$ w.r.t. π_j and λ to 0:

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial \pi_j} = \dfrac{\sum_{i=1}^{N^t} g_{ij}^{(m)}}{\pi_j} + \dfrac{\alpha_j - 1}{\pi_j} - \lambda = 0 \\ \dfrac{\partial \mathcal{L}}{\partial \lambda} = \sum\limits_{i=1}^{K} \pi_i - 1 = 0 \end{cases} \tag{A.15}$$

The solution to the above equation set can be written as:

$$\begin{cases} \pi_j = \dfrac{\sum_{i=1}^{N^t} g_{ij}^{(m)} + \alpha_j - 1}{\lambda} \\ \lambda = N + \sum\limits_{l=1}^{K} (\alpha_l - 1) \end{cases} \tag{A.16}$$

Therefore optimal $\boldsymbol{\pi}$ for $Q(\boldsymbol{\pi} | \boldsymbol{\pi}^{(m)})$ without inequality constraints is given by:

$$\pi_j = \frac{\sum_{i=1}^{N^t} g_{ij}^{(m)} + \alpha_j - 1}{N + \sum_{l=1}^{K} (\alpha_l - 1)} \tag{A.17}$$

Proof that the solution satisfies inequality constraints In the expression of $g_{ij}^{(m)}$ defined in Eq. (A.10), the output of the classifier $f : \mathcal{X} \to \Delta^{K-1}$ is a probability simplex and thus is non-negative. Note that we also have $c_i > 0, i = 1, 2...K$. Therefore $\boldsymbol{\pi}^{(m)} > 0 \Rightarrow g_{ij}^{(m)} > 0$ is non-negative. Note that we also require $\alpha_i - 1 > 0, i = 1, 2...K$ when defining the Dirichlet prior. Therefore we have $\boldsymbol{\pi}^{(m)} > 0 \Rightarrow \boldsymbol{\pi}^{(t+1)} > 0$.

Because the optimization problem is convex, when $\pi_j^{(m)} > 0, j = 1, 2, ...K$, the above equation gives the global optimal $\boldsymbol{\pi}^{(t+1)}$:

$$\pi_j^{(t+1)} = \left(\underset{\boldsymbol{\pi} \in \Delta^{K-1}}{\arg\max} Q(\boldsymbol{\pi} | \boldsymbol{\pi}^{(m)}) \right)_j = \frac{\sum_{i=1}^{N^t} g_{ij}^{(m)} + \alpha_j - 1}{N + \sum_{l=1}^{K} (\alpha_l - 1)} \tag{A.18}$$

□

Algorithm 9 MAPLS (Formal EM Formulation)

Input: Target domain unlabeled data $\{x_i^t | i = 1, 2, ...N, \{x_i^t, \cdot\} \sim p_t(x, y)\}$, source domain label distribution $p_s(y = j) = c_j$ and blackbox classifier $f : \mathcal{X} \to \Delta^{K-1}$, Dirichlet prior $p(\boldsymbol{\pi} | \boldsymbol{\alpha})$.

Initialize: $\boldsymbol{\pi}^{(0)} \in \Delta^{K-1}$ with $\pi_i^{(0)} > 0, i = 1, 2...K$

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

Estimating latent conditional distribution $g_{ij}^{(m)} := p_t(y = j | x_i^t, \boldsymbol{\pi}^{(m)})$:

$$g_{ij}^{(m)} = \frac{\frac{\pi_j^{(m)}}{c_j} f(x_i^t)_j}{\sum_{l=1}^{K} \frac{\pi_l^{(m)}}{c_l} f(x_i^t)_l} \tag{A.19}$$

E-step Construct $Q(\boldsymbol{\pi} | \boldsymbol{\pi}^{(m)})$ as:

$$\begin{aligned} Q(\boldsymbol{\pi} | \boldsymbol{\pi}^{(m)}) &= \underset{\mathbb{Y} | \mathcal{D}^t, \boldsymbol{\pi}^{(m)}}{\mathbb{E}} \left[\log p(\boldsymbol{\pi} | \mathcal{D}^t, \mathbb{Y}, \boldsymbol{\alpha}) \right] \\ &= \sum_{i=1}^{N^t} \sum_{j=1}^{K} g_{ij}^{(m)} \log \pi_j + \sum_{l=1}^{K} (\alpha_l - 1) \log \pi_l + Const \end{aligned} \tag{A.20}$$

M-step Maximize $Q(\boldsymbol{\pi} | \boldsymbol{\pi}^{(m)})$ w.r.t. $\boldsymbol{\pi} \in \Delta^{K-1}$:

$$\pi_j^{(t+1)} = \left(\underset{\boldsymbol{\pi} \in \Delta^{K-1}}{\arg\max} Q(\boldsymbol{\pi} | \boldsymbol{\pi}^{(m)}) \right)_j = \frac{\sum_{i=1}^{N^t} g_{ij}^{(m)} + \alpha_j - 1}{N + \sum_{l=1}^{K} (\alpha_l - 1)} \tag{A.21}$$

end for

Output: $p_t(y = \cdot) = \boldsymbol{\pi}^{(T+1)}$

A.2 Detailed Experimental Setups

A.2.1 Classifiers Details

We implement the Neural Netwrok classifier models using PyTorch (Paszke et al., 2017). We train a ResNet32 (Idelbayev, n.d.) classifier for CIFAR100 and every CIFAR100-LT dataset with weight decay $5e^{-4}$ for 200 epochs. The learning rate is initialized at 0.1 and drops by a factor of 10 at epochs 100 and 150. For ImageNet and Places, we use the pre-trained ResNet50 and ResNet152 respectively a classifier. For ImageNet-LT and Places-LT, we train a ResNet50 (He, Zhang et al., 2016) classifier with weight decay $2e^{-4}$ for 100 epochs. The learning rate is initialized to 0.1 and drops by a factor of 10 at epochs 60 and 80.

Dataset	Model	Setup	lr	weight decay	epoch	scheduler	mixup α
CIFAR100/100-LT	ResNet32	Scratch	0.1	$5e^{-4}$	200	lr decay 0.1 at $[100, 150]$	0.2
ImageNet	ResNet50	ImageNet Pre-Trained	-	-	-	-	-
ImageNet-LT	ResNet50	Scratch	0.1	$2e^{-4}$	100	lr decay 0.1 at $[60, 80]$	0.2
Places	ResNet152	Places Pre-Trained	-	-	-	-	-
Places-LT	ResNet152	ImageNet Pre-Trained	0.001	$1e^{-4}$	100	lr decay 0.1 at $[60, 80]$	0.1

Table A.1: Neural Network classifier setup used in our model.

For all the models training from scratch, we apply MixUp (Zhang, Cisse et al., 2018) with the parameter set to 0.2 during training. This is because MixUp is known to help increase Neural Network classifiers' calibration performance (Thulasidasan et al., 2019) and MLLS works better when classifier is calibrated on the source domain (Alexandari et al., 2020; Garg, Wu, Balakrishnan et al., 2020).

A.2.2 Label Shift Estimation Models Details

We report the performance of previous methods based on the source code below. MLLS code is provided by Alexandari et al. (2020) which have included the source code of RLLS (Azizzadenesheli et al., 2018) and BBSE (Lipton et al., 2018) with their original github page provided in the Tab. A.2. Only RLLS has hyperparameter in their model. We follow Alexandari et al. (2020) and RLLS original implementation to set the hyperparameter to be $\alpha = 0.01$.

Model Name	Source Code	Date of Retrieval
MLLS	https://github.com/kundajelab/labelshiftexperiments	Aug 2022
	https://github.com/kundajelab/abstention	Aug 2022
BBSE	https://github.com/flaviovdf/label-shift	Aug 2022
RLLS	https://github.com/Angie-Liu/labelshift	Aug 2022

Table A.2: Source Code details of reproduced existing label shift estimation models.

Note that confusion matrix based method like BBSE requires a validation set from source domain P_s to construct $K \times K$ confusion matrix. However, this is not feasible for dataset with larget K. For ImageNet, this means estimating $1000 \times 1000 = 1e^6$ elements in confusion matrix with only $5e^4$ validation samples. Therefore in our experiments, we use train set data to estimating the $K \times K$ confusion matrix. In ImageNet case, the $1e^6$ element in confusion matrix is then estimated with $1.28e^6$ samples rather than just $5e^4$ samples. This approach may introduce extra error, but is more feasible in practice (Lipton et al., 2018).

A.2.3 Experiment Setup Details

Label Shift Estimation Error: For label shift estimation error $(w - \hat{w})^2/K$, BBSE and RLLS are able to directly output $\hat{w}$ as a prediction to ground truth $w = p_t(y = \cdot)/p_s(y = \cdot)$. Therefore their performance can be computed directly. MLLS and our MAPLS/MAPLS-APL model predicts ground truth $\boldsymbol{\pi}$ with $\hat{\boldsymbol{\pi}}$. Thus we follow MLLS to compute $\hat{w} = \hat{\boldsymbol{\pi}}/\boldsymbol{c}$, where $\boldsymbol{c}$ is the source label distribution estimated by MLE given source domain labeled data.

Top1 Accuracy: The Top1 Accuracy of each label shift estimation model is obtained by first estimating the label shift with corresponding model, then correct label shift on target domain for classifier f, with offline label shift correction method defined in Eq. Eq. (2.34). We also report Top1 Accuracy on baseline classifier without any label shift correction.

Train and Test Sets: We test our MAPLS/MAPLS-APL model with as many different label shift settings as we can. Our experiments includes all the train-test set combinations of a train set in Tab. A.3 and a test set in Tab. A.4.

Dataset	Setup	Imbalance Ratio	Data Size	# of Classes	Top class sample	Tail class sample
CIFAR100	Original	None	50k	100	500	500
	Long-Tailed	2	36.0k	100	500	250
	Long-Tailed	5	24.8k	100	500	100
	Long-Tailed	10	19.5k	100	500	50
	Long-Tailed	20	15.9k	100	500	25
	Long-Tailed	50	12.6k	100	500	10
	Long-Tailed	100	10.8k	100	500	5
	Long-Tailed	200	9.5k	100	500	2
Places	Original	None	1803.4k	365	5000	3068
	Long-Tailed	996	62.5k	365	4980	5
ImageNet	Original	None	1281.1k	1000	1300	732
	Long-Tailed	256	115.8k	1000	1280	5

Table A.3: Detailed information of the training datasets with different label shift tested in Chapter 3.

Target Shift	Parameters
Original (Uniform)	None
Ordered Long-Tail (Hong, Han et al., 2021)	$R = \{2, 5, 10, 50\}$, Order = "Forward", "Backward"
Shuffled Long-Tail	$R = \{2, 5, 10, 50\}$
Dirichlet (Lipton et al., 2018)	$\alpha = 1.0, 10$

Table A.4: Detailed information of test datasets with different label shift in Chapter 3.

Appendix B

Appendix for Chapter 4

B.1 Mathematical Proofs

B.1.1 Proof of Proposition 3 (See page 64)

Proposition 3. ***(MLE)*** *Under Assumption 1, 4, if* $p_s(z|y=i) = q(\cdot|i,\theta)$*, then Eq. (4.9) is convex on* $\boldsymbol{\pi}$ *and EM algorithm 3 converges to a* $\boldsymbol{\pi}^{MLE}$ *defined in Eq. (4.10).*

Algorithm 10 GLSE-MLE

Input: $\mathcal{D}_z^t = \{z_i^t\}_{i=1}^{N^t}, q(\cdot|\cdot,\theta) \in \mathcal{Q}$.
Initialize: $\boldsymbol{\pi}^{(0)} \in \Delta_{>0}^{K-1}$.
for $t = 0$ to M **do**
E-step: Evaluate

$$g_{ij}^{(m)} = \frac{\pi_j^{(m)} q(z_i^t|j,\theta)}{\sum_{l=1}^{K} \pi_l^{(m)} q(z_i^t|l,\theta)}. \tag{4.11}$$

M-step: Evaluate

$$\pi_j^{(m+1)} = \frac{1}{N^t} \sum_{i=1}^{N^t} g_{ij}^{(m)}. \tag{4.12}$$

end for
Output: $p_t(y=\cdot) = \boldsymbol{\pi}^{(m+1)}$.

Proof. **Convexity of Negative Log Likelihood in Eq. (4.9)**

Recall that under A.1 and A.2, the negative log likelihood defined in Eq. (4.9) can be written as:

$$\begin{aligned} -\log L(\boldsymbol{\pi}; \mathcal{D}_z^t) &= -\log \prod_{i=1}^{N^t} \sum_{j=1}^{K} p_s(z_i^t|y=j) p_t(y=j;\boldsymbol{\pi}) \\ &= -\log \prod_{i=1}^{N^t} \sum_{j=1}^{K} q(z_i^t|j,\theta)\pi_j = -\sum_{i=1}^{N^t} \log \sum_{j=1}^{K} q(z_i^t|j,\theta)\pi_j. \end{aligned} \tag{B.1}$$

For any image $x_i \in \mathcal{D}^t$ and class $j \in \mathcal{Y}$, the term $q(z_i^t|j,\theta)$ is constant w.r.t. $\boldsymbol{\pi}$. Thus the negative log likelihood we consider is a sum of logarithm of linear combination of π_j. Note that $\boldsymbol{\pi} \in \Delta^{K-1}$ is defined in a convex set, therefore Eq. (4.9) is convex in $\boldsymbol{\pi}$.

□

Proof. **EM algorithm for MLE**

We follow the standard EM derivation procedure. The deviation has four steps:

1. Revisit the latent variable model that is considered, where $Y_t \sim \text{Cat}(K, \boldsymbol{\pi})$ is the latent variable.
2. Derive the form of complete likelihood $L(\boldsymbol{\pi}; \mathcal{D}_z^t, \mathbb{Y})$ for EM.
3. Construct $Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(m)})$ function for E-Step in EM.
4. Optimize $Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(m)})$ w.r.t. $\boldsymbol{\pi}$ to find the M-Step.

Step 1: Recall that in Eq. (4.3), under label shift we can construct latent variable model:

$$p_t(z) = \sum_{i=1}^{K} p_t(z|y=i)p_t(y=i) = \sum_{i=1}^{K} p_s(z|y=i)p_t(y=i). \tag{B.2}$$

In the above equation Eq. (B.2) we have:

- $Z_t \sim p_t(z)$ is the observed variable with samples available in the target unlabeled dataset $\mathcal{D}_z^t = \{f(x_i)|(x_i,\cdot) \sim p_t(x,y), i = 1,2,...,N\}$,
- Under label shift, $p_t(z|y=i) = p_s(z|y=i)$ is the class conditional distribution with available model $q(z|y,\theta)$,
- $Y_t \sim p_t(y) = \text{Cat}(K, \boldsymbol{\pi})$ is the unobserved latent variable.

Step 2: With Y_t as latent variable, let $\mathbb{Y} = \{y_i^t\}_{i=1}^{N^t}$ with $y_i^t \in \mathcal{Y}$, the complete likelihood $L(\boldsymbol{\pi}; \mathcal{D}_z^t, \mathbb{Y})$ can be written as:

$$\begin{aligned} L(\boldsymbol{\pi}; \mathcal{D}_z^t, \mathbb{Y}) &= \prod_{i=1}^{N}\prod_{j=1}^{K} p_t(z_i^t, y_i^t = j; \boldsymbol{\pi}) \\ &= \prod_{i=1}^{N}\prod_{j=1}^{K} p_t(z_i^t|y_i^t = j)p_t(y_i^t = j; \boldsymbol{\pi}) \\ &= \prod_{i=1}^{N}\prod_{j=1}^{K} q(z_i^t|j,\theta)\pi_j^{\mathbb{I}_j(y_i^t)}. \end{aligned} \tag{B.3}$$

Step 3: With the complete likelihood $L(\boldsymbol{\pi}; \mathcal{D}_z^t, \mathbb{Y})$, we can construct the $Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(m)})$ in the **E-Step** as:

$$\begin{aligned} Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(m)}) &= \mathop{\mathbb{E}}_{\mathbb{Y}|\mathcal{D}_z^t, \boldsymbol{\pi}^{(m)}} \left[\log L(\boldsymbol{\pi}; \mathcal{D}_z^t, \mathbb{Y})\right] \\ &= \mathop{\mathbb{E}}_{\mathbb{Y}|\mathcal{D}_z^t, \boldsymbol{\pi}^{(m)}} \left[\log \prod_{i=1}^{N} \prod_{j=1}^{K} q(z_i^t|j,\theta) \pi_j^{\mathbb{I}_j(y_i^t)}\right] \\ &= \mathop{\mathbb{E}}_{\mathbb{Y}|\mathcal{D}^t, \boldsymbol{\pi}^{(m)}} \left[\sum_{i=1}^{N} \sum_{j=1}^{K} \mathbb{I}_j(y_i^t) \log \pi_j + Const\right] \\ &= \sum_{i=1}^{N} \sum_{j=1}^{K} p_t(y_i^t = j|z_i^t; \boldsymbol{\pi}^{(m)}) \log \pi_j + Const \\ &= \sum_{i=1}^{N} \sum_{j=1}^{K} g_{ij}^{(m)} \log \pi_j + Const, \end{aligned} \tag{B.4}$$

where the likelihood $g_{ij}^{(m)} := p_t(y_i^t = j|z_i^t; \boldsymbol{\pi}^{(m)})$ can be simply obtained via:

$$g_{ij}^{(m)} = \frac{q(z_i^t|j,\theta)\pi_j^{(m)}}{\sum_{l=1}^{K} q(z_i^t|l,\theta)\pi_l^{(m)}}. \tag{B.5}$$

Step 4: In the **M-step**, we solve the optimization objective with respect to $\boldsymbol{\pi}$:

$$\boldsymbol{\pi}^{(m+1)} = \mathop{\arg\max}_{\boldsymbol{\pi} \in \Delta^{K-1}} Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(m)}). \tag{B.6}$$

By substitution, the objective can be rewritten as:

$$\begin{cases} \min\limits_{\boldsymbol{\pi}} - \sum\limits_{i=1}^{N^t} \sum\limits_{j=1}^{K} g_{ij}^{(m)} \log \pi_j \\ \text{s.t: } \sum\limits_{j=1}^{K} \pi_j = 1 \text{ and } \pi_i \geq 0, i \in [1,2,...K]. \end{cases} \tag{B.7}$$

Convexity The objective we want to optimize is just a linear combination of $\log \pi_i$, which is a concave function w.r.t. $\boldsymbol{\pi}$. Recall that $\boldsymbol{\pi} \in \Delta^{K-1}$ is a convex set on $\mathbb{R}^K$, the above optimization problem is a convex optimization problem and every local minima is a global minima.

Optimization without inequality constraints With only equality constraints, the standard Lagrangian Multiplier method can be applied. The Lagrangian can be written as:

$$\mathcal{L}(\boldsymbol{\pi}, \lambda) = \sum_{i=1}^{N^t} \sum_{j=1}^{K} g_{ij}^{(m)} \log \pi_j + \lambda \left(1 - \sum_{j=1}^{K} \pi_j\right). \tag{B.8}$$

The optimal $\boldsymbol{\pi}$ can be found by taking all the partial derivatives of $\mathcal{L}(\boldsymbol{\pi}, \lambda)$ w.r.t. π_j and λ to 0:

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial \pi_j} = \dfrac{\sum_{i=1}^{N^t} g_{ij}^{(m)}}{\pi_j} - \lambda = 0 \\ \dfrac{\partial \mathcal{L}}{\partial \lambda} = \sum\limits_{i=1}^{K} \pi_i - 1 = 0. \end{cases} \tag{B.9}$$

The solution to the above equation set can be written as:

$$\begin{cases} \pi_j = \dfrac{\sum_{i=1}^{N^t} g_{ij}^{(m)}}{\lambda} \\ \lambda = N. \end{cases} \tag{B.10}$$

Therefore optimal $\boldsymbol{\pi}$ for $Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(m)})$ without inequality constraints is given by:

$$\pi_j = \frac{\sum_{i=1}^{N^t} g_{ij}^{(m)}}{N}. \tag{B.11}$$

Proof that the solution satisfies inequality constraints π_j in Eq. (B.11) depends on $g_{ij}^{(m)}$ defined in Eq. (B.5), where $q(z|j,\theta) \in \mathbb{R}_{\geq 0}$ is the evaluation of a probability density function. Therefore $\boldsymbol{\pi}^{(m)} > 0 \Rightarrow \boldsymbol{\pi}^{(m+1)} \geq 0$. Because the optimization problem is convex, if we initialize $\pi_j^{(0)} \in \mathbb{R}_{>0}^K$, Eq. (B.11) gives the global optimal $\boldsymbol{\pi}^{(m+1)}$ that satisfies:

$$\pi_j^{(m+1)} = \left(\underset{\boldsymbol{\pi} \in \Delta^{K-1}}{\arg\max} Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(m)}) \right)_j = \frac{\sum_{i=1}^{N^t} g_{ij}^{(m)}}{N}. \tag{B.12}$$

□

B.1.2 Proof of Lemma 4 (See page 65)

Lemma 4. *If the $p(\boldsymbol{\pi}|\boldsymbol{\alpha})$ in Eq. (4.13) is log strictly concave, then objective Eq. (4.13) is strictly convex on $\boldsymbol{\pi}$.*

Proof. Recall that the negative log posterior defined in Eq. (4.13) can be written as:

$$\begin{aligned} -\log p(\boldsymbol{\pi}|\mathcal{D}_z^t, \boldsymbol{\alpha}) &= -\log p(\boldsymbol{\pi}|\boldsymbol{\alpha}) \prod_{i=1}^{N^t} \sum_{j=1}^{K} q(z_i^t|j,\theta)\pi_j \\ &= -\log p(\boldsymbol{\pi}|\boldsymbol{\alpha}) - \log L(\boldsymbol{\pi}; \mathcal{D}_z^t), \end{aligned} \tag{B.13}$$

where $\log L(\boldsymbol{\pi}; \mathcal{D}_z^t)$ is the log likelihood.

As proved in Section B.1.1, the negative log likelihood $-\log L(\boldsymbol{\pi}; \mathcal{D}_z^t)$ is convex on $\boldsymbol{\pi}$. If $p(\boldsymbol{\pi}|\boldsymbol{\alpha})$ is log strictly concave, then $-\log p(\boldsymbol{\pi}|\boldsymbol{\alpha})$ is strictly convex. The overall objective is the sum of the two terms and thus is strictly convex on $\boldsymbol{\pi}$.

□

B.1.3 Proof of Proposition 5 (See page 65)

Proposition 5. ***(MAP estimate)*** *Under Assumption 1, 4 with* $p_s(z|y=i) = q(\cdot|i,\theta)$. *If parameter* $\boldsymbol{\pi} \sim Dir(K, \boldsymbol{\alpha})$ *with* $\boldsymbol{\alpha} \in \mathbb{R}^K_{>1}$, *then EM algorithm 4 converges to the* $\boldsymbol{\pi}^{MAP}$ *defined in Eq.* (4.14).

Algorithm 11 GLSE-MAP

Input: $\mathcal{D}_z^t = \{z_i^t\}_{i=1}^{N^t}$, $q(\cdot|\cdot,\theta) \in \mathcal{Q}$, $\boldsymbol{\alpha} \in \mathbb{R}^K_{>1}$.
Initialize: $\boldsymbol{\pi}^{(0)} \in \Delta^{K-1}_{>0}$.
for $t = 0$ to M **do**
E-step: Evaluate

$$g_{ij}^{(m)} = \frac{\pi_j^{(m)} q(z_i^t|j,\theta)}{\sum_{l=1}^K \pi_l^{(m)} q(z_i^t|l,\theta)}. \tag{4.15}$$

M-step: Evaluate

$$\pi_j^{(m+1)} = \frac{\sum_{i=1}^{N^t} g_{ij}^{(m)} + \alpha_j - 1}{N^t + \sum_{l=1}^K (\alpha_l - 1)}. \tag{4.16}$$

end for
Output: $p_t(y=\cdot) = \boldsymbol{\pi}^{(m+1)}$.

Proof. The deviation is similar to EM algorithm for MLE derivation in Appendix B.1.1 and also has four steps:

1. Revisit the latent variable model that is considered, where $Y_t \sim p_t(y) = \text{Cat}(K, \boldsymbol{\pi})$ is the latent variable.
2. Derive the form of complete posterior $P(\boldsymbol{\pi}; \mathcal{D}_z^t, \mathbb{Y}, \boldsymbol{\alpha})$ for EM.
3. Construct $Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(m)})$ function for E-Step in EM.
4. Optimize $Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(m)})$ w.r.t. $\boldsymbol{\pi}$ to find the M-Step.

Step 1: Recall that in Eq. (4.3), under label shift we can construct latent variable model:

$$p_t(z) = \sum_{i=1}^K p_t(z|y=i)p_t(y=i) = \sum_{i=1}^K p_s(z|y=i)p_t(y=i). \tag{B.14}$$

In the above Eq. (B.14) we have:

- $Z_t \sim p_t(z)$ is the observed variable with samples available in the target unlabeled dataset $\mathcal{D}_z^t = \{f(x_i)|(x_i,\cdot) \sim p_t(x,y), i = 1,2,...,N\}$,
- Under label shift, $p_t(z|y=i) = p_s(z|y=i)$ is the class conditional distribution with available model $q(z|y,\theta)$,
- $Y_t \sim p_t(y) = \text{Cat}(K,\boldsymbol{\pi})$ is the unobserved latent variable.

Step 2: With Y_t as latent variable, let $\mathbb{Y} = \{y_i^t\}_{i=1}^{N^t}$ with $y_i^t \in \mathcal{Y}$, the complete posterior $P(\boldsymbol{\pi}|\mathcal{D}_z^t, \mathbb{Y}, \boldsymbol{\alpha})$ give Dirichlet prior $\boldsymbol{\pi} \sim \text{Dir}(K,\boldsymbol{\alpha})$ ($\boldsymbol{\alpha} \in \mathbb{R}_{>1}^K$) can be written as:

$$\begin{aligned} p(\boldsymbol{\pi}|\mathcal{D}_z^t, \mathbb{Y}, \boldsymbol{\alpha}) &= \frac{1}{Const} p(\boldsymbol{\pi}|\boldsymbol{\alpha}) \prod_{i=1}^{N}\prod_{j=1}^{K} p_t(z_i^t, y_i^t = j; \boldsymbol{\pi}) \\ &= \frac{1}{Const} p(\boldsymbol{\pi}|\boldsymbol{\alpha}) \prod_{i=1}^{N}\prod_{j=1}^{K} p_t(z_i^t|y_i = j) p_t(y_i = j; \boldsymbol{\pi}) \\ &= \frac{1}{Const} \prod_{l=1}^{K} \pi_l^{\alpha_l - 1} \prod_{i=1}^{N}\prod_{j=1}^{K} q(z_i^t|j,\theta) \pi_j^{\mathbb{I}_j(y_i^t)}, \end{aligned} \tag{B.15}$$

where $p(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \frac{1}{\text{B}(\boldsymbol{\alpha})} \prod_{i=1}^{K} \pi_i^{\alpha_i - 1}$ is the Dirichlet prior and $Const$ includes all the terms that are constant w.r.t. $\boldsymbol{\pi}$.

Step 3: Given the complete posterior $p(\boldsymbol{\pi}|\mathcal{D}_z^t, \mathbb{Y}, \boldsymbol{\alpha})$ defined in Eq. (B.15), we can construct the $Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(m)})$ in the **E-Step** as:

$$\begin{aligned} Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(m)}) &= \mathop{\mathbb{E}}_{\mathbb{Y}|\mathcal{D}_z^t, \boldsymbol{\pi}^{(m)}} \left[\log P(\boldsymbol{\pi}|\mathcal{D}_z^t, \mathbb{Y}, \boldsymbol{\alpha})\right] \\ &= \mathop{\mathbb{E}}_{\mathbb{Y}|\mathcal{D}_z^t, \boldsymbol{\pi}^{(m)}} \left[\log P(\boldsymbol{\pi}|\boldsymbol{\alpha}) \prod_{i=1}^{N}\prod_{j=1}^{K} q(z_i^t|j,\theta) \pi_j^{\mathbb{I}_j(y_i^t)}\right] \\ &= \mathop{\mathbb{E}}_{\mathbb{Y}|\mathcal{D}^t, \boldsymbol{\pi}^{(m)}} \left[\sum_{i=1}^{N}\sum_{j=1}^{K} \mathbb{I}_j(y_i^t) \log \pi_j + \sum_{l=1}^{K} (\alpha_l - 1) \log \pi_l + Const\right] \\ &= \sum_{i=1}^{N}\sum_{j=1}^{K} p_t(y_i^t = j|z_i^t; \boldsymbol{\pi}^{(m)}) \log \pi_j + \sum_{l=1}^{K} (\alpha_l - 1) \log \pi_l + Const \\ &= \sum_{i=1}^{N}\sum_{j=1}^{K} g_{ij}^{(m)} \log \pi_j + \sum_{l=1}^{K} (\alpha_l - 1) \log \pi_l + Const, \end{aligned} \tag{B.16}$$

where the likelihood $g_{ij}^{(m)} := p_t(y_i^t = j|z_i^t; \boldsymbol{\pi}^{(m)})$ can be simply obtained via:

$$g_{ij}^{(m)} = \frac{q(z_i^t|j,\theta)\pi_j^{(m)}}{\sum_{l=1}^{K} q(z_i^t|l,\theta)\pi_l^{(m)}}. \tag{B.17}$$

Step 4: With available $Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(m)})$, we solve the optimization objective with respect to $\boldsymbol{\pi}$ in the **M-step**, :

$$\boldsymbol{\pi}^{(m+1)} = \arg\max_{\boldsymbol{\pi}\in\Delta^{K-1}} Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(m)}). \tag{B.18}$$

By substitution, the objective can be rewritten as:

$$\begin{cases} \min\limits_{\boldsymbol{\pi}} -\sum\limits_{i=1}^{N^t}\sum\limits_{j=1}^{K} g_{ij}^{(m)} \log \pi_j - \sum\limits_{l=1}^{K}(\alpha_l - 1)\log \pi_l \\ \text{s.t: } \sum\limits_{j=1}^{K} \pi_j = 1 \text{ and } \pi_i \geq 0, i \in [1, 2, ...K]. \end{cases} \tag{B.19}$$

Convexity The objective we want to optimize is just a linear combination of $\log \pi_i$, which is a concave function w.r.t. $\boldsymbol{\pi}$. Knowing that the constraints define a convex set on $\mathbb{R}^K$, the above optimization problem is also a convex optimization problem and every local minima is a global minima.

Optimization without inequality constraints With only equality constraints, standard the Lagrangian Multiplier method can be applied. The Lagrangian can be written as:

$$\mathcal{L}(\boldsymbol{\pi}, \lambda) = \sum_{i=1}^{N^t}\sum_{j=1}^{K} g_{ij}^{(m)} \log \pi_j + \sum_{j=1}^{K}(\alpha_j - 1)\log \pi_j + \lambda\left(1 - \sum_{j=1}^{K}\pi_j\right). \tag{B.20}$$

The optimal $\boldsymbol{\pi}$ can be found by taking all the partial derivative of $\mathcal{L}(\boldsymbol{\pi}, \lambda)$ w.r.t. π_j and λ to 0:

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial \pi_j} = \dfrac{\sum_{i=1}^{N^t} g_{ij}^{(m)}}{\pi_j} + \dfrac{\alpha_j - 1}{\pi_j} - \lambda = 0 \\ \dfrac{\partial \mathcal{L}}{\partial \lambda} = \sum\limits_{i=1}^{K} \pi_i - 1 = 0. \end{cases} \tag{B.21}$$

The solution to the above equation set can be written as:

$$\begin{cases} \pi_j = \dfrac{\sum_{i=1}^{N^t} g_{ij}^{(m)} + \alpha_j - 1}{\lambda} \\ \lambda = N^t + \sum\limits_{l=1}^{K}(\alpha_l - 1). \end{cases} \tag{B.22}$$

Therefore optimal $\boldsymbol{\pi}$ for $Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(m)})$ without inequality constraints is given by:

$$\pi_j = \frac{\sum_{i=1}^{N^t} g_{ij}^{(m)} + \alpha_j - 1}{N^t + \sum_{l=1}^{K}(\alpha_l - 1)}. \tag{B.23}$$

Proof that the solution satisfies inequality constraints

$g_{ij}^{(m)}$ defined in Eq. (B.17), where $q(z|j,\theta) \in \mathbb{R}_{\geq 0}$ is the evaluation of a probability density function. Therefore $\boldsymbol{\pi}^{(m)} > 0 \Rightarrow \boldsymbol{\pi}^{(m+1)} \geq 0$. Because the optimization problem is convex, if we initialize $\pi_j^{(0)} \in \mathbb{R}_{>0}^K$, Eq. (B.11) gives the global optimal $\boldsymbol{\pi}^{(m+1)}$ that satisfies:

$$\pi_j^{(m+1)} = \left(\underset{\boldsymbol{\pi} \in \Delta^{K-1}}{\arg\max} Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(m)}) \right)_j = \frac{\sum_{i=1}^{N^t} g_{ij}^{(m)} + \alpha_j - 1}{N^t + \sum_{l=1}^{K}(\alpha_l - 1)}. \tag{B.24}$$

□

B.1.4 Proof of Proposition 6 (See page 66)

Proposition 6. *Under Assumption 1 and Assumption 4, let $p_s(y=i|z) = h(i;\cdot)$ and $Y_s \sim Cat(K, \boldsymbol{c})$, if we define:*

$$q(z_0|i,\theta) = h(i;z_0)/\theta_i \tag{4.19}$$

with $\theta := \boldsymbol{c} \in \Delta^{K-1}$, then GLSE-MLE 3 converges to a $\boldsymbol{\pi}^{MLE}$ and GLSE-MAP 4 converges to the unique $\boldsymbol{\pi}^{MAP}$.

Proof. For a feature extractor $f : \mathcal{X} \to \mathcal{Z}$ with $p_s(y|z) = h(y;z)$ and $Y_s \sim \text{Cat}(K, \boldsymbol{c})$, $p_s(z|y=i)$ by Bayes rule as:

$$p_s(z|y=i) = \frac{p_s(y=i|z)p_s(z)}{p_s(y=i)} = \frac{h(z;i)}{c_i} p_s(z), \tag{B.25}$$

where $p_s(z)$ is the marginal distribution of feature z on source domain.

Because $p_s(z)$ is constant w.r.t. $\boldsymbol{\pi}$, MLE/MAP estimate of $\boldsymbol{\pi}$ can still be obtained without knowing $p_s(z)$. Specifically, in the E-step of Algorithm 3 and 4 we have:

$$\begin{aligned} g_{ij}^{(m)} &= \frac{\pi_j^{(m)} q(z_i^t|j,\theta)}{\sum_{l=1}^{K} \pi_l^{(m)} q(f(x_i)|l,\theta)} \\ &= \frac{\pi_j^{(m)} h(f(x_i);j) p_s(z)/c_j}{\sum_{l=1}^{K} \pi_l^{(m)} h(f(x_i);l) p_s(z)/c_l} \\ &= \frac{\pi_j^{(m)} h(f(x_i);j)/c_j}{\sum_{l=1}^{K} \pi_l^{(m)} h(f(x_i);l)/c_l}. \end{aligned} \tag{B.26}$$

The $p_s(z)$ cancelled out and we are still able to obtain $\boldsymbol{\pi}^{\text{MLE}}, \boldsymbol{\pi}^{\text{MAP}}$ by viewing $q(z_0|i,\theta) = h(i;z_0)/\theta_i$ with $\theta_j := c_j$.

□

B.1.5 Derivation of Example 7 (See page 68)

Example 7. *For $f : \mathcal{X} \to \{1, 2, ..., L\}$, under Assumption 1, Assumption 4 with $\mathcal{Q}$ defined in Eq. (4.4), Alg. 5 converges to the $\boldsymbol{\pi}^{MAP}$.*

Proof. When $\mathcal{Z} = \{1, 2, ..., L\}$ is discrete, based on definition of $\mathcal{Q}$ in Eq. (4.4) we have:

$$q(z|j, \theta) = \theta^j_z. \tag{B.27}$$

Substituting into GLSE-MLE algorithm 3, the **E-Step** becomes:

$$g^{(m)}_{ij} = \frac{q(z^t_i|j, \theta)\pi^{(m)}_j}{\sum_{l=1}^K q(f(x_i)|l, \theta)\pi^{(m)}_l} = \frac{\theta^j_{f(x_i)}\pi^{(m)}_j}{\sum_{l=1}^K \theta^l_{f(x_i)}\pi^{(m)}_l}. \tag{B.28}$$

Substituting the result of **E-Step** into the **M-Step** we have:

$$\pi^{(m+1)}_j = \frac{1}{N^t}\sum_{i=1}^{N^t} g^{(m)}_{ij} = \frac{1}{N^t}\sum_{i=1}^{N^t} \frac{\theta^j_{f(x_i)}\pi^{(m)}_j}{\sum_{l=1}^K \theta^l_{f(x_i)}\pi^{(m)}_l} = \sum_{i=1}^{L} \frac{b_i\theta^j_i\pi^{(m)}_j}{\sum_{l=1}^K \theta^l_i\pi^{(m)}_l}, \tag{B.29}$$

where $b_i = \frac{1}{N^t}\sum_{j=1}^{N^t} \mathbb{I}_i(f(x_j))$.

Similarly, for GLSE-MAP 4 that compute $\boldsymbol{\pi}^{\text{MAP}}$, we have the same **E-Step** as GSLE-MLE 3. Based on Eq. (4.17), the GLSE-MAP algorithm can be rewritten as:

$$\begin{aligned} \pi^{(m+1)}_j &= \lambda\frac{\sum_{i=1}^{N^t} g^{(m)}_{ij}}{N} + (1-\lambda)\frac{\alpha_j - 1}{\sum_{l=1}^K \alpha_l - 1} \\ &= \lambda\sum_{i=1}^{L} \frac{b_i\theta^j_i\pi^{(m)}_j}{\sum_{l=1}^K \theta^l_i\pi^{(m)}_l} + (1-\lambda)\frac{\alpha_j - 1}{\sum_{l=1}^K (\alpha_l - 1)}, \end{aligned} \tag{B.30}$$

where $\lambda = N^t/(N^t + \sum_{l=1}^K(\alpha_l - 1))$ and $b_i = \frac{1}{N^t}\sum_{j=1}^{N^t} \mathbb{I}_i(f(x_j))$.

□

B.1.6 Full Likelihood Estimation of θ

Under our GLSE model, we have the label shift assumption (Assumption 1) and distribution family assumption (Assumption 4):

A.1 $p_s(z|y = i) = p_t(z|y = i)$ for all $i \in \mathcal{Y}$

A.2 $p_s(z|y) \in \mathcal{Q} = \{q(\cdot|\cdot, \theta)|\theta \in \Theta\}$

In the label shift estimation problem setup, we are provided with a feature extractor $f : \mathcal{X} \to \mathcal{Z}$, labeled source domain sample $\mathcal{D}^s = \{(x^s_i, y^s_i)\}_{i=1}^{N^s}$ and unlabeled target domain sample $\mathcal{D}^t = \{x_i\}_{i=1}^{N^t}$. Based on this information, we can obtain:

- Source domain samples of $Z_s|Y_s = i$ in $\mathcal{D}^s_z = \{(f(x^s_i), y^s_i)|(x^s_i, y^s_i) \in \mathcal{D}^s\}$.

- Target domain samples of Z_t in $\mathcal{D}_z^t = \{f(x_i)|x_i \in \mathcal{D}^t\}$.
- $Y_t \sim \text{Cat}(K, \boldsymbol{\pi})$.

The latent variable model (LVM) can be constructed by viewing Y_t as latent variable and Z_t as observed variable:

$$p_t(z) = \sum_{i=1}^{K} p_t(z|y=i)p_t(y=i) = \sum_{i=1}^{K} p_s(z|y=i)p_t(y=i), \tag{B.31}$$

The likelihood of θ and $\boldsymbol{\pi}$ given $\mathcal{D}_z^s, \mathcal{D}_z^t$ can be written as:

$$L(\theta, \boldsymbol{\pi}; \mathcal{D}_z^s, \mathcal{D}_z^t) = \prod_{i=1}^{N^s} q(f(x_i^s)|y_i^s, \theta) \prod_{j=1}^{N^t} \sum_{l=1}^{K} q(f(x_j)|l, \theta)\pi_l. \tag{B.32}$$

The parameter $\boldsymbol{\pi}, \theta$ can be estimated by maximizing the likelihood $L(\theta, \boldsymbol{\pi}; \mathcal{D}_z^s, \mathcal{D}_z^t)$. Any MLE of θ satisfies:

$$\begin{aligned} \theta^{\text{MLE}} &\in -\arg\min_{\theta\in\Theta} \log L(\theta, \boldsymbol{\pi}; \mathcal{D}_z^s, \mathcal{D}_z^t) \\ &= -\arg\min_{\theta\in\Theta} \left(\sum_{i=1}^{N^s} \log q(f(x_i^s)|y_i^s, \theta) + \sum_{j=1}^{N^t} \log \sum_{l=1}^{K} q(f(x_j)|l, \theta)\pi_l \right), \end{aligned} \tag{B.33}$$

which is different to our GLSE framework, where θ^{MLE} is obtained with only source domain data. In fact, our GLSE model only considers the first term in Eq. (B.33).

And any $\boldsymbol{\pi}^{\text{MLE}}$ satisfies:

$$\begin{aligned} \boldsymbol{\pi}^{\text{MLE}} &\in -\arg\min_{\boldsymbol{\pi}\in\Delta^{K-1}} \log L(\theta, \boldsymbol{\pi}; \mathcal{D}_z^s, \mathcal{D}_z^t) \\ &= -\arg\min_{\boldsymbol{\pi}\in\Delta^{K-1}} \left(\sum_{j=1}^{N^t} \log \sum_{l=1}^{K} q(f(x_j)|l, \theta)\pi_l \right), \end{aligned} \tag{B.34}$$

which is identical to Eq. (4.10) that obtain $\boldsymbol{\pi}^{\text{MLE}}$ in our GLSE framework.

In our GLSE framework, we propose to obtain θ^{MLE} with source domain data $\mathcal{D}^s$ based on the following reasons:

1. **Simplicity:** Maximizing the likelihood in Eq. (B.32) w.r.t. $\boldsymbol{\pi}$ and θ together is more complicated.
2. **Efficiency:** Under our GLSE framework, θ^{MLE} is estimated only once with source domain $\mathcal{D}_z^s$ and then free to use for any target domain dataset $\mathcal{D}_z^t$. Whereas for Eq. (B.33), θ^{MLE} have to be re-estimated for every single $\mathcal{D}_z^t$.
3. **Practical Reason:** In practice, it usually happens that $N^s >> N$. In this case, the first term in Eq. (B.34) dominates and the θ^{MLE} obtained by Eq. (4.10) approximate the θ^{MLE} obtained by Eq. (B.34) well.

B.2 Detailed Experimental Setups

B.2.1 Classifiers Details

We implement the Neural Network classifier models using PyTorch (Paszke et al., 2017). We train a ResNet32 (Idelbayev, n.d.) classifier for CIFAR100 and every CIFAR100-LT dataset with weight decay $5e^{-4}$ for 200 epochs. The learning rate is initialized at 0.1 and drops by a factor of 10 at epochs 100 and 150. For ImageNet and Places, we use the pre-trained ResNet50 and ResNet152 respectively a classifier. For ImageNet-LT and Places-LT, we train a ResNet50 (He, Zhang et al., 2016) classifier with weight decay $2e^{-4}$ for 100 epochs. The learning rate is initialized to 0.1 and drops by a factor of 10 at epochs 60 and 80.

Dataset	Model	Train Setup	lr	weight decay	epoch	scheduler	mixup α
CIFAR10/100-LT	ResNet32	from scratch	0.1	$5e^{-4}$	200	lr decay 0.1 at $[100, 150]$	0.2
ImageNet	ResNet50	ImageNet pre-trained	-	-	-	-	-
ImageNet-LT	ResNet50	from scratch	0.1	$2e^{-4}$	100	lr decay 0.1 at $[60, 80]$	0.2
Places	ResNet152	Places pre-trained	-	-	-	-	-
Places-LT	ResNet152	ImageNet pre-trained	0.001	$1e^{-4}$	100	lr decay 0.1 at $[60, 80]$	0.1

Table B.1: Neural Network classifier setup used in our model.

For all the models training from scratch, we apply MixUp (Zhang, Cisse et al., 2018) with the parameter set to 0.2 during training. This is because MixUp is known to help increase Neural Network classifiers' calibration performance (Thulasidasan et al., 2019) and MLLS works better when classifier is calibrated on the source domain (Alexandari et al., 2020; Garg, Wu, Balakrishnan et al., 2020).

B.2.2 Label Shift Estimation Models Details

We report the performance of previous methods based on the source code below. MLLS code is provided by Alexandari et al. (2020) which includes the source code of RLLS (Azizzadenesheli et al., 2018) and BBSE (Lipton et al., 2018) with their original github page provided in the Tab. B.2. Only RLLS has hyperparameters in their model. We follow Alexandari et al. (2020) and the RLLS original implementation to set the hyperparameter to be $\alpha = 0.01$.

Model Name	Source Code	Date of Retrieval
MLLS	`https://github.com/kundajelab/labelshiftexperiments`	Aug 2022
	`https://github.com/kundajelab/abstention`	Aug 2022
BBSE	`https://github.com/flaviovdf/label-shift`	Aug 2022
RLLS	`https://github.com/Angie-Liu/labelshift`	Aug 2022

Table B.2: Source Code details of reproduced existing label shift estimation models.

For MLLS and our proposed model, we follow MLLS (Alexandari et al., 2020) to initialize target label distribution the same as source domain label distribution $\boldsymbol{\pi}^{(0)} = \boldsymbol{c}$. Each EM has $T = 100$ iterations to get the final estimation.

Note that confusion matrix based method like BBSE require a validation set from the source domain $p_s(x, y)$ to construct $K \times K$ confusion matrix. However, this

is not feasible for datasets with larget K. For ImageNet, this means estimating $1000 \times 1000 = 1e^6$ elements in confusion matrix with only $5e^4$ validation samples. Therefore in our experiments, we use train set data to estimate the $K \times K$ confusion matrix. In the ImageNet case, the $1e^6$ element in confusion matrix is then estimated with $1.28e^6$ samples rather than just $5e^4$ samples. This approach may introduce extra error, but is more feasible in practice (Lipton et al., 2018).

B.2.3 Datasets and Evaluation Metrics Details

Label Shift Estimation Error: For label shift estimation error $(w - \hat{w})^2/K$, BBSE and RLLS are able to directly output $\hat{w}$ as a prediction to ground truth $w_i = p_t(y = i)/p_s(y = i)$. Therefore their performance can be computed directly. MLLS and our GLSE$_{z/c}$/GLSE$_{z/c}$-APL model predict ground truth $\boldsymbol{\pi}$ with $\hat{\boldsymbol{\pi}}$. Thus we follow MLLS to compute $\hat{w} = \hat{\boldsymbol{\pi}}/\boldsymbol{c}$, where $\boldsymbol{c}$ is the source label distribution estimated by MLE given source domain labeled data.

Top1 Accuracy: The Top1 Accuracy of each label shift estimation model is obtained by first estimating the label shift with corresponding model, then correct label shift on target domain for classifier f, with offline label shift correction method defined in Eq. Eq. (2.34). We also report Top1 Accuracy on baseline classifier without any label shift correction.

Train and Test Sets: We test our GLSE$_{z/c}$/GLSE$_{z/c}$-APL model with as many different label shift settings as we can. Our experiments includes all the train-test set combinations of a train set in Tab. B.3 and a test set in Tab. B.4.

Dataset	Setup	Imbalance Ratio	Data Size	# of Classes	Top class sample	Tail class sample
CIFAR100	Original	None	50k	100	500	500
	Long-tailed	10	19.5k	100	500	50
	Long-tailed	100	10.8k	100	500	5
Places	Original	None	1803.4k	365	5000	3068
	Long-tailed	996	62.5k	365	4980	5
ImageNet	Original	None	1281.1k	1000	1300	732
	Long-tailed	256	115.8k	1000	1280	5

Table B.3: Detailed information of train sets with different label shift in Chapter 4.

Target Shift	Parameters
Original (Uniform)	None
Ordered Long-Tail (Hong, Han et al., 2021)	$R = \{2, 5, 10, 25, 50\}$, Order = "Forward", "Backward"
Shuffled Long-Tail	$R = \{2, 5, 10, 25, 50\}$
Dirichlet (Lipton et al., 2018)	$\alpha = 1.0, 10$

Table B.4: Detailed information of test sets with different label shift in Chapter 4.

Appendix C

Appendix for Chapter 5

C.1 Mathematical Proofs

C.1.1 Proof of Theorem 8 (See page 84)

Theorem 8. *(**Source ID/OOD ratio estimator**) Under Assumption 5B, given source ID dataset $\mathcal{D}^s$ and source OOD dataset $\mathcal{D}^o$, then for all $\delta > 0$, with probability of at least $1 - 2\delta$,*

$$|\rho_s - \hat{\rho}_s| \leq \frac{1}{1 - \mu_1 + \mu_0} \sqrt{\frac{\log 1/\delta}{2 \min(|\mathcal{D}^o|, |\mathcal{D}^s|)}} \tag{5.4}$$

where

$$\mu_0 := \mathbb{E}_{X_s|B_s=0}[h(x)] \quad \textit{and} \quad \mu_1 := \mathbb{E}_{X_s|B_s=1}[h(x)]. \tag{5.5}$$

Proof. Given the available information, for $p_s(b=1) = \rho_s$ we have:

$$\begin{aligned} p_s(b=1) &= \mathbb{E}_{X_s}[p(b=1|x)] = \mathbb{E}_{X_s}[h(x)] = \mathbb{E}_{B_s}[\mathbb{E}_{X_s|B_s}[h(x)]] \\ &= (1 - p_s(b=1)) \cdot \mathbb{E}_{X_s|B_s=0}[h(x)] + p_s(b=1) \cdot \mathbb{E}_{X_s|B_s=1}[h(x)] \end{aligned} \tag{C.1}$$

Rearranging the equation and we can get:

$$\begin{aligned} \rho_s &= \frac{\mathbb{E}_{X_s|B_s=0}[h(x)]}{1 - \mathbb{E}_{X_s|B_s=1}[h(x)] + \mathbb{E}_{X_s|B_s=0}[h(x)]} \\ &= \frac{\mu_0}{1 - \mu_1 + \mu_0}, \end{aligned} \tag{C.2}$$

where $\mu_0 := \mathbb{E}_{X_s|B_s=0}[h(x)]$ and $\mu_1 := \mathbb{E}_{X_s|B_s=1}[h(x)]$.

The expectation terms can be approximated given OOD dataset $\mathcal{D}^{\mathbf{o}}$ and source ID dataset $\mathcal{D}^s$:

$$\begin{cases} \mathbb{E}_{X_s|B_s=0}[h(x)] \approx \dfrac{1}{|\mathcal{D}^{\mathbf{o}}|} \displaystyle\sum_{x \in \mathcal{D}^{\mathbf{o}}} h(x) \\ \mathbb{E}_{X_s|B_s=1}[h(x)] \approx \dfrac{1}{|\mathcal{D}^s|} \displaystyle\sum_{x \in \mathcal{D}^s} h(x), \end{cases} \tag{C.3}$$

which yields the approximation $\hat{\rho}$:

$$\hat{\rho} = \frac{\hat{\mu}_0}{1 - \hat{\mu}_1 + \hat{\mu}_0}, \tag{C.4}$$

where $\hat{\mu}_0 := \frac{1}{|\mathcal{D}^{\mathbf{0}}|} \sum_{x \in \mathcal{D}^{\mathbf{0}}} h(x)$ and $\hat{\mu}_1 := \frac{1}{|\mathcal{D}^s|} \sum_{x \in \mathcal{D}^s} h(x)$.

Note that since $h(x) \in [0,1]$, the Hoeffding's inequality (Vershynin, 2018) guarantees for all $\epsilon > 0$:

$$\begin{aligned} p\left(|\mu_0 - \hat{\mu}_0| \geq \epsilon\right) &\leq 2e^{-2|\mathcal{D}^{\mathbf{0}}|\epsilon^2} \\ p\left(|\mu_1 - \hat{\mu}_1| \geq \epsilon\right) &\leq 2e^{-2|\mathcal{D}^s|\epsilon^2}. \end{aligned} \tag{C.5}$$

Therefore with high probability of at least $1 - 2e^{-2\min(|\mathcal{D}^{\mathbf{0}}|,|\mathcal{D}^s|)\epsilon^2}$ we have:

$$\begin{cases} \rho - \hat{\rho} \leq \dfrac{\mu_0}{1 - \mu_1 + \mu_0} - \dfrac{\mu_0 + \epsilon}{1 - (\mu_1 + \epsilon) + (\mu_0 + \epsilon)} = \dfrac{\epsilon}{1 - \mu_1 + \mu_0}, \\ \rho - \hat{\rho} \geq \dfrac{\mu_0}{1 - \mu_1 + \mu_0} - \dfrac{\mu_0 - \epsilon}{1 - (\mu_1 - \epsilon) + (\mu_0 - \epsilon)} = \dfrac{-\epsilon}{1 - \mu_1 + \mu_0}, \end{cases} \tag{C.6}$$

for all $\delta \in [0, \max((1-\mu_0)/2, (1-\mu_1)/2)]$, which is equivalent to:

$$|\rho - \hat{\rho}| < \frac{\epsilon}{1 - \mu_1 + \mu_0}. \tag{C.7}$$

Letting $\delta := e^{-2\min(|\mathcal{D}^{\mathbf{0}}|,|\mathcal{D}^s|)\epsilon^2}$, rearrange the equations and we get the result. □

C.1.2 Extension of Theorem 8 to the Multi-Class setting (See page 84)

Problem Setup: (General) Given a blackbox model $h : \mathcal{X} \to \Delta^{K-1}$ that satisfies $h(x) = p(y|x)$ for distribution $p(x,y)$ and K datasets $\mathcal{D}^1, \mathcal{D}^2, ..., \mathcal{D}^K$, with $\mathcal{D}^k$ containing samples drawn i.i.d. from $p(x|y=k)$, we want to estimate the label distribution $p(y=\cdot) = \boldsymbol{\rho} \in \Delta^{K-1}$.

Similar to the binary case, we can write the label distribution as the sum of the conditional expectation:

$$\begin{aligned} \rho_j &= \mathbb{E}_X[p(y=j|x)] = \mathbb{E}_X[h(x)_j] = \mathbb{E}_Y[\mathbb{E}_{X|Y}[h(x)_j]] \\ &= \sum_{k=1}^{K} p(y=k)\mathbb{E}_{X|Y}[h(x)_j] = (\mu\boldsymbol{\rho})_j, \end{aligned} \tag{C.8}$$

where $\mu \in \mathbb{R}^{K \times K}$ with $\mu_{jk} := \mathbb{E}_{X|Y=k}[h(x)_j]$.

Lemma 12. ***(Multi-Class)*** *If $p(y|x) = h(x)$, given $\mathcal{D}^1, \mathcal{D}^2, ..., \mathcal{D}^K$ containing samples x drawn i.i.d. from $p(x|y=1), p(x|y=2), ..., p(x|y=K)$ for K classes,*

then for $p(y) = \boldsymbol{\rho} \in \Delta^{K-1}$ *we have:*

$$\arg\min_{\boldsymbol{\rho}\in\Delta^{K-1}} \|(\hat{\mu} - \mathbf{I})\boldsymbol{\rho}\|_2^2 \xrightarrow[a.s.]{} \boldsymbol{\rho}, \tag{C.9}$$

where $\hat{\mu} \in \mathbb{R}^{K\times K}$ *is a stochastic matrix with* $\mu_{jk} = \frac{1}{|\mathcal{D}^k|}\sum_{x\in\mathcal{D}^k} h(x)_j$.

Proof. Given the available information, let $p(y = j) = \rho_j$ for all $j \in \mathcal{Y} = \{1, 2, ..., K\}$, then we have:

$$\begin{aligned}\rho_j &= \mathbb{E}_X[p(y = j|x)] = \mathbb{E}_X[h(x)_j] = \mathbb{E}_Y[\mathbb{E}_{X|Y}[h(x)_j]] \\ &= \sum_{k=1}^{K} p(y = k)\mathbb{E}_{X|Y}[h(x)_j] \\ &= (\mu\boldsymbol{\rho})_j,\end{aligned} \tag{C.10}$$

where $\mu \in \mathbb{R}^{K\times K}$ with $\mu_{jk} := \mathbb{E}_{X|Y=k}[h(x)_j]$.

The μ can be approximated via:

$$\mu_{jk} \approx \hat{\mu}_{jk} := \frac{1}{|\mathcal{D}^k|}\sum_{x\in\mathcal{D}^k} h(x)_j. \tag{C.11}$$

Hence we can approximate $\boldsymbol{\rho}$ with $\hat{\boldsymbol{\rho}}$ that is defined as:

$$\hat{\boldsymbol{\rho}} := \arg\min_{\boldsymbol{\rho}\in\Delta^{K-1}} \|(\hat{\mu} - \mathbf{I})\boldsymbol{\rho}\|_2^2. \tag{C.12}$$

□

C.1.3 Proof of Lemma 9 (See page 86)

Lemma 9. *Under Assumption 1 and Assumption 5, given* $\mathcal{D}^t$*, the negative log likelihood of parameter* $\boldsymbol{\pi}$ *and* ρ_t *can be written as:*

$$-\log L(\boldsymbol{\pi}, \rho_t; \mathcal{D}^t) = -\sum_{i=1}^{N^t} \log\left(\sum_{j=1}^{K+1} \frac{\tilde{\pi}_j}{\tilde{c}_j}\tilde{f}(x_i)_j\right) + C, \tag{5.6}$$

where C *does not depend on either* $\boldsymbol{\pi}$ *or* ρ_t *and*

$$\tilde{f}(x)_i := \begin{cases} h(x)\cdot f(x)_i, & i \in \mathcal{Y} \\ 1 - h(x), & i = K+1, \end{cases} \tag{5.7}$$

$$\begin{aligned}\tilde{\boldsymbol{\pi}} &:= [\rho_t\cdot\pi_1, ..., \rho_t\cdot\pi_K, 1-\rho_t]^T \\ \tilde{\boldsymbol{c}} &:= [\rho_s\cdot c_1, ..., \rho_s\cdot c_K, 1-\rho_s]^T.\end{aligned} \tag{5.8}$$

Proof. The label shift assumption can be written as:

$$p_s(x|y=i) = p_t(x|y=i) \quad \text{for all} \quad i \in \mathcal{Y} \cup \{K+1\} \tag{C.13}$$

On target domain, if we are given only unlabeled images $\mathcal{D}^t = \{x_i^t\}_{i=1}^{N^t}$, we can construct the likelihood:

$$\begin{aligned} L(\boldsymbol{\pi}, \rho_t; \mathcal{D}^t) &= \prod_{i=1}^{N^t} p_t(x; \boldsymbol{\pi}, \rho_t) \\ &= \prod_{i=1}^{N^t} \left(\sum_{l=1}^{2} \sum_{j=1}^{K} p_t(x_i|y=j) p_t(y=j|b=l) p_t(b=l) \right). \end{aligned} \tag{C.14}$$

Note that $p_s(b=1) = \rho_s$, $p_t(b=1) = \rho_t$ and in Eq. (5.1) for all $(x, j) \in \mathcal{X} \times (\mathcal{Y} \cup \{K+1\})$ we have:

$$\begin{aligned} p_s(y|b; \boldsymbol{c}) &= \begin{cases} c_j, & \text{if } b=1, y \in \mathcal{Y} \\ 1, & \text{if } b=0, y=K+1 \\ 0, & \text{otherwise,} \end{cases} \\ p_s(y|b; \boldsymbol{c}) &= \begin{cases} \pi_j, & \text{if } b=1, y \in \mathcal{Y} \\ 1, & \text{if } b=0, y=K+1 \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{C.15}$$

Based on Eq. (C.15) and label shift Assumption 1 we have:

$$
\begin{aligned}
L(\boldsymbol{\pi}, \rho_t; \mathcal{D}^t) =& \prod_{i=1}^{N^t} \left(\sum_{l=1}^{2} \sum_{j=1}^{K} p_t(x_i|y=j) p_t(y=j|b=l) p_t(b=l) \right) \\
=& \prod_{i=1}^{N^t} \Bigg(\sum_{j=1}^{K} p_t(x_i|y=j) p_t(y=j|b=1) p_t(b=1) + \\
& p_t(x_i|y=K+1) p_t(y=K+1|b=0) p_t(b=0) \Bigg) \\
=& \prod_{i=1}^{N^t} \Bigg(\sum_{j=1}^{K} p_s(x_i|y=j) p_t(y=j|b=1) p_t(b=1) + \\
& p_s(x_i|y=K+1) p_t(y=K+1|b=0) p_t(b=0) \Bigg) \\
=& \prod_{i=1}^{N^t} \Bigg(\sum_{j=1}^{K} \frac{p_s(y=j|x_i)}{p_s(y=j)} p_t(y=j|b=1) p_t(b=1) + \\
& \frac{p_s(y=K+1|x_i)}{p_s(y=K+1)} p_t(y=K+1|b=0) p_t(b=0) \Bigg) \cdot Const,
\end{aligned}
\tag{C.16}
$$

where $Const := \prod_{i=1}^{N^t} p_s(x_i)$ is irrelevant to $\boldsymbol{\pi}$ or ρ_t.

Based on Eq. (C.15) and Assumption 5 we have $p_s(y = \cdot|x, b = 1) = f(x)$ and $p_s(b = 1|x) = h(x)$, therefore:

$$
\begin{aligned}
1 - h(x) &= p_s(b=0|x_i) \\
&= \sum_{i=1}^{K+1} p_s(b=0|y=i) p(y=i|x_i) \\
&= \sum_{i=1}^{K+1} \frac{p_s(y=i|b=0) p_s(b=0)}{p_s(y=i)} p(y=i|x_i) \\
&= \frac{1 \cdot p_s(b=0)}{\sum_{j=1}^{2} p_s(y=K+1|b=j) p_s(b=j)} p_s(y=K+1|x_i) \\
&= \frac{p_s(b=0)}{p_s(b=0)} \cdot p_s(y=K+1|x_i) = p_s(y=K+1|x_i),
\end{aligned}
\tag{C.17}
$$

and for $j \in \{1, 2, ..., K\}$ we have:

$$
\begin{aligned}
p_s(y=j|x_i) =& p_s(y=j|x_i, b=1) p_s(b=1|x_i) \\
&+ p_s(y=j|x_i, b=0) p_s(b=0|x_i) \\
=& f(x)_j \cdot \rho_s + 0 \cdot (1 - h(x)) = h(x) \cdot f(x_i)_j.
\end{aligned}
\tag{C.18}
$$

Marginalize Eq. (C.15) we can also get:

$$p_s(y=j)=\begin{cases}c_j\cdot\rho_s, & j\neq K+1\\ 1-\rho_s, & j=K+1\end{cases},\quad p_t(y=j)=\begin{cases}\pi_j\cdot\rho_t, & j\neq K+1\\ 1-\rho_t, & j=K+1\end{cases}, \tag{C.19}$$

Substituting Eq. (C.15) and Eq. (C.19) into the likelihood Eq. (C.16) we get:

$$\begin{aligned}L(\boldsymbol{\pi},\rho_t;\mathcal{D}^t)&=\prod_{i=1}^{N^t}\left(\sum_{j=1}^{K}\frac{h(x)\cdot f(x_i)_j}{\rho_s\cdot c_j}\pi_j\cdot\rho_t+\frac{1-h(x)}{1-\rho_s}\cdot 1\cdot(1-\rho_t)\right)\cdot Const,\\&=\prod_{i=1}^{N^t}\left(\frac{\rho_t}{\rho_s}h(x_i)\cdot\sum_{j=1}^{K}\frac{\pi_j}{c_j}f(x_i)_j+\frac{1-\rho_t}{1-\rho_s}\cdot(1-h(x_i))\right)\cdot Const.\end{aligned} \tag{C.20}$$

Further substitute Eq. (5.7) and Eq. (5.8) into Eq. (C.20) and then we can get the result.

□

C.1.4 Proof of Theorem 10 (See page 87)

Theorem 10. ***(MLE)*** *Under Assumption 1 and Assumption 5, the the NLL (5.6) is convex in $\tilde{\boldsymbol{\pi}}$ (and convex in ρ_t), and the EM algorithm MLE-OLS (Alg. 7) converges to $\boldsymbol{\pi}^{MLE},\rho_t^{MLE}$ (5.9).*

Algorithm 12 MLE-OLS

Input: $\mathcal{D}_f^t=\{x_i^t\}_{i=1}^{N^t},\boldsymbol{c},\rho_s,h(x),f(x)$.

Initialize: $\boldsymbol{\pi}^{(0)}\in\Delta_{>0}^{K-1},\rho_t^{(0)}\in(0,1)$

for $m=0$ to M **do**

Construct: $\tilde{\boldsymbol{\pi}}^{(m)}$ based on $\boldsymbol{\pi}^{(m)},\rho_t^{(m)}$ and Eq. (5.8).

E-step: For $j\in\mathcal{Y}\cup\{K+1\}$, evaluate

$$g_{ij}^{(m)}=\frac{\tilde{\pi}_j^{(m)}/\tilde{c}_j\cdot\tilde{f}(x_i^t)_j}{\sum_{l=1}^{K}\tilde{\pi}_l^{(m)}/\tilde{c}_l\cdot\tilde{f}(x_i^t)_l}. \tag{C.21}$$

M-step: For $j\in\mathcal{Y}$, evaluate

$$\begin{cases}\pi_j^{(m+1)}=\dfrac{\sum_{i=1}^{N^t}g_{ij}^{(m)}}{N-\sum_{i=1}^{N^t}g_{iK+1}^{(m)}}\\ \rho_t^{(m+1)}=\dfrac{N-\sum_{i=1}^{N^t}g_{iK+1}^{(m)}}{N}\end{cases} \tag{C.22}$$

end for

Output: $p_t(y=\cdot)=\boldsymbol{\pi}^{(M+1)},p_t(b=1)=\rho_t^{(M+1)}$.

Proof. **Convexity**: As shown in Lemma 9 Eq. (C.20), the negative log likelihood of $\boldsymbol{\pi}, \rho_t$ given Assumption 1,5 and unlabeled target domain dataset $\mathcal{D}^t$ can be written as:

$$\begin{aligned} -\log L(\boldsymbol{\pi}, \rho_t; \mathcal{D}^t) = -\sum_{i=1}^{N^t} \log \Bigg(\frac{\rho_t}{\rho_s} h(x_i) \cdot \sum_{j=1}^{K} \frac{\pi_j}{c_j} f(x_i)_j \\ + \frac{1-\rho_t}{1-\rho_s} \cdot (1 - h(x_i)) \Bigg) + Const. \end{aligned} \tag{C.23}$$

As a function of ρ_t, the NLL can be rewritten as:

$$-\log L(\boldsymbol{\pi}, \rho_t; \mathcal{D}^t) = -\sum_{i=1}^{N^t} \log(A\rho_t + B) + Const, \tag{C.24}$$

which is a convex function w.r.t. ρ_t.

As a function of $\boldsymbol{\pi}$, the NLL can be rewritten as:

$$-\log L(\boldsymbol{\pi}, \rho_t; \mathcal{D}^t) = -\sum_{i=1}^{N^t} \log \left(A \sum_{j=1}^{K} \frac{\pi_j}{c_j} f(x_i)_j + B \right) + Const, \tag{C.25}$$

which is a convex function w.r.t. $\boldsymbol{\pi}$.

Moreover, same as the close world setting (Alexandari et al., 2020), the NLL is convex in the reparameterisation of $\tilde{c}_t$. □

Proof. **EM algorithm**: The NLL objective of MLE defined in Lemma 9, Eq. (5.6) can be rewritten as:

$$-\log L(\boldsymbol{\pi}, \rho_t; \mathcal{D}^t) = -\sum_{i=1}^{N^t} \log \left(\sum_{j=1}^{K} \frac{\tilde{\pi}_j}{\tilde{c}_j} \tilde{f}(x_i)_j \right) + Const, \tag{C.26}$$

which is reparametrised as the objective of the closed set label shift estimation model MLLS (Saerens et al., 2002) algorithm (Chapter 2, Eq. (2.32)).

As MLE is invariant under reparametrisation (Murphy, 2012), and MLLS have been proved to converge to a MLE estimate (Alexandari et al., 2020), thus EM algorithm 12 converge to a $\tilde{c}_t^{\text{MLE}}$ and will also converge to a $\boldsymbol{\pi}^{\text{MLE}}, \rho_t^{\text{MLE}}$.

The MLE can be seen as a special case of MAP estimate with prior distribution being 1. In this case, by setting $\boldsymbol{\alpha}^{\mathbf{in}} = \mathbf{1}, \alpha_1^{\mathbf{out}} = 1, \alpha_2^{\mathbf{out}}) = 1$. Proof of EM algorithm for MAP estimate can be found in Proof of Proposition 13. □

C.1.5 MAP estimation of target label distribution parameters

MAP estimate: Moreover, if we employ a prior $\boldsymbol{\pi} \sim p(\boldsymbol{\pi}|\boldsymbol{\alpha}^{\mathbf{in}})$ over the target label distribution $\boldsymbol{\pi}$, or a prior $\rho_t \sim p(\rho_t|\boldsymbol{\alpha}^{\mathbf{out}})$ over the target ID data ratio ρ_t, we can

Algorithm 13 MAP-OLS

Input: $\mathcal{D}_f^t = \{x_i^t\}_{i=1}^{N^t}, \boldsymbol{c}, \rho_s, h(x), f(x), \boldsymbol{\alpha}^{\text{in}}, \alpha_1^{\text{out}}, \alpha_2^{\text{out}}$.
Require: $\boldsymbol{\alpha}^{\text{in}} \in \mathbb{R}_{>1}^K, \alpha_1^{\text{out}}, \alpha_2^{\text{out}} \in \mathbb{R}_{>1}$.
Initialize: $\boldsymbol{\pi}^{(0)} \in \Delta_{>0}^{K-1}, \rho_t^{(0)} \in (0,1)$.
Construct: $\tilde{f}$ based on Eq. (5.7).
for $m = 0$ to M **do**
Construct: $\tilde{\boldsymbol{\pi}}^{(m)}$ based on $\boldsymbol{\pi}^{(m)}, \rho_t^{(m)}$ and Eq. (5.8).
E-step: For $j \in \mathcal{Y} \cup \{K+1\}$, evaluate

$$g_{ij}^{(m)} = \frac{\tilde{\pi}_j^{(m)} / \tilde{c}_j \cdot \tilde{f}(x_i^t)_j}{\sum_{l=1}^K \tilde{\pi}_l^{(m)} / \tilde{c}_l \cdot \tilde{f}(x_i^t)_l}. \tag{C.29}$$

M-step: For $j \in \mathcal{Y}$, evaluate

$$\begin{cases} \pi_j^{(m+1)} = \dfrac{\sum_{i=1}^{N^t} g_{ij}^{(m)} + \alpha_j^{\text{in}} - 1}{N - \sum_{i=1}^{N^t} g_{iK+1}^{(m)} + \sum_{l=1}^K (\alpha_l^{\text{in}} - 1)} \\ \rho_t^{(m+1)} = \dfrac{N - \sum_{i=1}^{N^t} g_{iK+1}^{(m)} + \alpha_1^{\text{out}} - 1}{N^t + \alpha_1^{\text{out}} + \alpha_2^{\text{out}} - 2}. \end{cases} \tag{C.30}$$

end for
Output: $p_t(y = \cdot) = \boldsymbol{\pi}^{(M+1)}, p_t(b = 1) = \rho_t^{(M+1)}$.

construct the posterior of $\boldsymbol{\pi}$ and ρ_t as:

$$\begin{aligned} -\log p(\boldsymbol{\pi}, \rho_t | \mathcal{D}^t, \boldsymbol{\alpha}) = &-\log L(\boldsymbol{\pi}, \rho_t; \mathcal{D}^t) - \log p(\boldsymbol{\pi} | \boldsymbol{\alpha}^{\text{in}}) \\ &- \log p(\rho_t | \boldsymbol{\alpha}^{\text{out}}) + Const. \end{aligned} \tag{C.27}$$

In this work, inspired by Ye et al. (2024), we show that with a Dirichlet prior over $\boldsymbol{\pi} \sim \text{Dir}(K, \boldsymbol{\alpha}^{\text{in}})$ or a Beta prior over $\rho_t \sim \text{Beta}(\alpha_1^{\text{out}}, \alpha_2^{\text{out}})$, the MAP estimate $\tilde{\boldsymbol{\pi}}^{\text{MAP}}$ can be obtained via another EM algorithm over the objective:

$$\boldsymbol{\pi}^{\text{MAP}}, \rho_t^{\text{MAP}} \in \arg\min_{\tilde{\pi} \in \Delta^K} -\log p(\boldsymbol{\pi}, \rho_t | \mathcal{D}^t, \boldsymbol{\alpha}), \tag{C.28}$$

where the details are also provided in Proposition 13.

Proposition 13. ***(MAP)*** *Under Assumption 1, 5, if* $\boldsymbol{\pi} \sim Dir(K, \boldsymbol{\alpha}^{in})$ *with* $\boldsymbol{\alpha}^{in} \in \mathbb{R}_{>1}^K$ *and* $\rho_t \sim Beta(\alpha_1^{out}, \alpha_2^{out})$ *with* $\alpha_1^{out}, \alpha_2^{out} \in \mathbb{R}_{>1}$, *then*

- *The posterior in Eq. (C.27) is strictly convex in* $\boldsymbol{\pi}$ *and strictly convex in* ρ_t.
- *EM algorithm 7 converge to the* $\tilde{\boldsymbol{\pi}}^{MAP}$ *in Eq. (C.28).*

Proof. **Convexity**: As shown in the Proof Proposition 10, the MLE objective given in Lemma 9 is convex on $\boldsymbol{\pi}, \rho_t$.

Since Dirichlet prior $\boldsymbol{\pi} \sim \text{Dir}(K, \boldsymbol{\alpha}^{\text{in}})$ with $\boldsymbol{\alpha}^{\text{in}} > \mathbf{1}$ is strictly convex on $\boldsymbol{\pi}$. And Beta prior $\rho_t \sim \text{Beta}(\alpha_1^{\text{out}}, \alpha_2^{\text{out}})$ with $\alpha_1^{\text{out}}, \alpha_2^{\text{out}} > 1$ is strictly convex on ρ_t, the overall

posterior:

$$\begin{aligned} -\log p(\boldsymbol{\pi}, \rho_t | \mathcal{D}^t, \boldsymbol{\alpha}) = &-\log L(\boldsymbol{\pi}, \rho_t; \mathcal{D}^t) - \log p(\boldsymbol{\pi} | \boldsymbol{\alpha}^{\mathbf{in}}) \\ &- \log p(\rho_t | \boldsymbol{\alpha}^{\mathbf{out}}) + Const \end{aligned} \tag{C.31}$$

is strictly convex on $\boldsymbol{\pi}$ and ρ_t

□

Proof. **EM algorithm:**

To be concise, we will use the notation:

$$\begin{aligned} \tilde{f}(x)_i &= \begin{cases} h(x) \cdot f(x)_i, & i \in \mathcal{Y} \\ 1 - h(x), & i = K+1, \end{cases} \\ \tilde{\boldsymbol{\pi}} &= [\rho_t \cdot \pi_1, ..., \rho_t \cdot \pi_K, 1 - \rho_t]^T \\ \tilde{\boldsymbol{c}} &= [\rho_s \cdot c_1, ..., \rho_t \cdot c_K, 1 - \rho_s]^T \end{aligned} \tag{C.32}$$

Remark: We proof the case with the model having both prior $\boldsymbol{\pi} \sim \text{Dir}(K, \boldsymbol{\alpha}^{\mathbf{in}})$ and $\rho_t \sim \text{Beta}(\alpha_1^{\mathbf{out}}, \alpha_2^{\mathbf{out}})$, where $\boldsymbol{\alpha}^{\mathbf{in}} \in \mathbb{R}^K_{>1}$ and $\alpha_1^{\mathbf{out}}, \alpha_2^{\mathbf{out}} \in \mathbb{R}_{>1}$. EM algorithms for other cases can be derived similarly by setting $\boldsymbol{\alpha}^{\mathbf{in}} = \mathbf{1}$ or $\alpha_1^{\mathbf{out}} = 1, \alpha_2^{\mathbf{out}} = 1$ or both.

The proof consists of three stages:

1. Identify the latent variable, derive the complete posterior;
2. Construct the $Q(\boldsymbol{\pi}, \rho_t | \boldsymbol{\pi}^{(m)}, \rho_t^{(m)})$ and obtain **E-Step**;
3. Optimize $Q(\boldsymbol{\pi}, \rho_t | \boldsymbol{\pi}^{(m)}, \rho_t^{(m)})$ w.r.t. $\boldsymbol{\pi}, \rho_t$ and obtain **M-Step**.

Step 1: As discussed in the main paper (Eq. (5.6)), we can construct the latent variable $\tilde{Y}_s \sim \text{Cat}(K+1, \tilde{\boldsymbol{c}})$ and $\tilde{Y}_t \sim \text{Cat}(K+1, \tilde{\boldsymbol{\pi}})$. With $\tilde{Y}_t$ as latent variable, let $\tilde{\mathbb{Y}} = \{\tilde{y}_i^t\}_{i=1}^{N^t}$ with $\tilde{y}_i^t \in \mathcal{Y} \cup \{K+1\}$, the complete posterior $p(\boldsymbol{\pi} | \mathcal{D}^t, \tilde{\mathbb{Y}}, \boldsymbol{\alpha}^{\mathbf{in}}, \alpha_1^{\mathbf{out}}, \alpha_2^{\mathbf{out}})$

can be written as:

$$
\begin{aligned}
p(\boldsymbol{\pi}, \rho_t | \mathcal{D}_f^t, \tilde{\mathbb{Y}}, \boldsymbol{\alpha}^{\mathbf{in}}, \alpha_1^{\mathbf{out}}, \alpha_2^{\mathbf{out}}) =& \frac{1}{C} p(\boldsymbol{\pi} | \boldsymbol{\alpha}^{\mathbf{in}}) \cdot p(\rho_t | \alpha_1^{\mathbf{out}}, \alpha_2^{\mathbf{out}}) \\
& \times \prod_{i=1}^{N} \prod_{j=1}^{K+1} p_t(x_i^t, \tilde{y}_i^t = j; \tilde{\boldsymbol{\pi}}) \\
=& \frac{1}{C} p(\boldsymbol{\pi} | \boldsymbol{\alpha}^{\mathbf{in}}) \cdot p(\rho_t | \alpha_1^{\mathbf{out}}, \alpha_2^{\mathbf{out}}) \\
& \times \prod_{i=1}^{N} \prod_{j=1}^{K+1} \frac{p_t(\tilde{y}_i^t = j; \tilde{\boldsymbol{\pi}})}{p_s(\tilde{y}_i^t = j)} p_s(\tilde{y}_i^t = j | x_i^t) \\
=& \frac{1}{C} \rho_t^{\alpha_1^{\mathbf{out}}} \cdot (1 - \rho_t)^{\alpha_2^{\mathbf{out}}} \cdot \prod_{l=1}^{K} \pi_l^{\alpha_l^{\mathbf{in}} - 1} \\
& \times \prod_{i=1}^{N} \prod_{j=1}^{K+1} \left(\frac{\tilde{\pi}_j}{\tilde{c}_j} \right)^{\mathbb{I}_j(\tilde{y}_i^t)} \tilde{f}(x_i^t)_j,
\end{aligned}
\tag{C.33}
$$

where C includes all the terms that are constant w.r.t. $\boldsymbol{\pi}, \rho_t$.

Step 2: Given the complete posterior $p(\boldsymbol{\pi} | \mathcal{D}_f^t, \tilde{\mathbb{Y}}, \boldsymbol{\alpha}^{\mathbf{in}}, \alpha_1^{\mathbf{out}}, \alpha_2^{\mathbf{out}})$, we can construct the $Q(\boldsymbol{\pi}, \rho_t | \boldsymbol{\pi}^{(m)}, \rho_t^{(m)})$ in the **E-Step** as:

$$
\begin{aligned}
Q(\boldsymbol{\pi}, \rho_t | \boldsymbol{\pi}^{(m)}, \rho_t^{(m)}) =& \mathop{\mathbb{E}}_{\tilde{\mathbb{Y}} | \mathcal{D}^t, \boldsymbol{\pi}^{(m)}, \rho_t^{(m)}} \left[\log p(\boldsymbol{\pi}, \rho_t | \mathcal{D}_f^t, \tilde{\mathbb{Y}}, \boldsymbol{\alpha}^{\mathbf{in}}, \alpha_1^{\mathbf{out}}, \alpha_2^{\mathbf{out}}) \right] \\
=& \mathop{\mathbb{E}}_{\tilde{\mathbb{Y}} | \mathcal{D}^t, \boldsymbol{\pi}^{(m)}, \rho_t^{(m)}} \Bigg[\sum_{i=1}^{N} \sum_{j=1}^{K+1} \mathbb{I}_j(\tilde{y}_i^t) \log \tilde{\pi}_j + \sum_{l=1}^{K} (\alpha_l - 1) \log \pi_l \\
& + \alpha_1^{\mathbf{out}} \cdot \log \rho_t + \alpha_2^{\mathbf{out}} \cdot \log(1 - \rho_t) + C \Bigg] \\
=& \sum_{i=1}^{N} \sum_{j=1}^{K+1} p_t(\tilde{y}_i^t = j | x_i^t; \tilde{\boldsymbol{\pi}}^{(m)}) \log \tilde{\pi}_j + \sum_{l=1}^{K} (\alpha_l - 1) \log \pi_l \\
& + \alpha_1^{\mathbf{out}} \cdot \log \rho_t + \alpha_2^{\mathbf{out}} \cdot \log(1 - \rho_t) + C \\
=& \sum_{i=1}^{N} \sum_{j=1}^{K+1} g_{ij}^{(m)} \log \tilde{\pi}_j + \sum_{l=1}^{K} (\alpha_l - 1) \log \pi_l \\
& + \alpha_1^{\mathbf{out}} \cdot \log \rho_t + \alpha_2^{\mathbf{out}} \cdot \log(1 - \rho_t) + C,
\end{aligned}
\tag{C.34}
$$

where the likelihood $g_{ij}^{(m)} := p_t(\tilde{y}_i^t = j | x_i^t; \boldsymbol{\pi}^{(m)}, \rho^{(m)})$ can be simply obtained via:

$$
g_{ij}^{(m)} = \frac{\frac{\tilde{\pi}_j^{(m)}}{\tilde{c}_j^{(m)}} \tilde{f}(x_i)_j}{\sum_{l=1}^{K+1} \frac{\tilde{\pi}_l^{(m)}}{\tilde{c}_l^{(m)}} \tilde{f}(x_i)_l} \quad \text{for all} \quad j \in \mathcal{Y} \cup \{K+1\}. \tag{C.35}
$$

Step 3: In the **M-step**, with available $Q(\boldsymbol{\pi}, \rho_t | \boldsymbol{\pi}^{(m)}, \rho_t^{(m)})$, we solve the optimization objective with respect to $\boldsymbol{\pi}$ by fixing ρ_t and vise versa:

$$\boldsymbol{\pi}^{(m+1)}, \rho_t^{(m+1)} = \underset{\boldsymbol{\pi} \in \Delta^{K-1}, \rho_t \in [0,1]}{\arg\max} \; Q(\boldsymbol{\pi}, \rho_t | \boldsymbol{\pi}^{(m)}, \rho_t^{(m)}) \tag{C.36}$$

By substitution, the objective can be rewritten as:

$$\begin{cases} \min\limits_{\boldsymbol{\pi}} - \sum\limits_{i=1}^{N^t} \sum\limits_{j=1}^{K+1} g_{ij}^{(m)} \log \tilde{\pi}_j - \sum\limits_{l=1}^{K} (\alpha_l^{\textbf{in}} - 1) \log \pi_l - \alpha_1^{\textbf{out}} \cdot \log \rho_t \\ \qquad - \alpha_2^{\textbf{out}} \cdot \log(1 - \rho_t) \\ \text{s.t: } \sum\limits_{j=1}^{K} \pi_j = 1, \tilde{\boldsymbol{\pi}} = [\rho_t \cdot \pi_1, ..., \rho_t \cdot \pi_K, 1 - \rho_t]^T, \\ \qquad \pi_i \geq 0 \text{ for } i \in [1, 2, ...K], \rho_t \in [0, 1]. \end{cases} \tag{C.37}$$

Convexity Eq. (C.37) is just a linear combination of $\log \pi_i$, which is a concave function w.r.t. $\boldsymbol{\pi}$. Knowing that the constraints define a convex set on $\mathbb{R}^K$, therefore Eq. (C.37) is convex w.r.t. $\boldsymbol{\pi}$ and every local minima is a global minima. Similarly, it's also easy to show that Eq. (C.37) is also convex w.r.t. ρ_t for $\rho_t \in [0, 1]$.

Optimization without inequality constraints With only equality constraints, standard the Lagrangian Multiplier method can be applied. The Lagrangian can be written as:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\pi}, \rho_t, \lambda) = & \sum_{i=1}^{N^t} \sum_{j=1}^{K} g_{ij}^{(m)} \log(\rho_t \cdot \pi_j) + \sum_{i=1}^{N^t} g_{iK+1}^{(m)} \log(1 - \rho_t) \\ & + \sum_{j=1}^{K} (\alpha_j^{\textbf{in}} - 1) \log \pi_j + (\alpha_1^{\textbf{out}} - 1) \cdot \log \rho_t \\ & + (\alpha_2^{\textbf{out}} - 1) \cdot \log(1 - \rho_t) + \lambda \left(1 - \sum_{j=1}^{K} \pi_j \right). \end{aligned} \tag{C.38}$$

The optimal $\boldsymbol{\pi}, \rho_t$ can be found by taking all the partial derivative of $\mathcal{L}(\boldsymbol{\pi}, \rho_t, \lambda)$ w.r.t. π_j, ρ_t and λ to 0:

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial \pi_j} = \dfrac{\sum_{i=1}^{N^t} g_{ij}^{(m)}}{\pi_j} + \dfrac{\alpha_j^{\textbf{in}} - 1}{\pi_j} - \lambda = 0 \\ \dfrac{\partial \mathcal{L}}{\partial \rho_t} = \dfrac{\sum_{i=1}^{N^t} \sum_{j=1}^{K} g_{ij}^{(m)} + (\alpha_1^{\textbf{out}} - 1)}{\rho_t} - \dfrac{\sum_{i=1}^{N^t} g_{iK+1}^{(m)} + (\alpha_2^{\textbf{out}} - 1)}{1 - \rho_t} = 0 \\ \dfrac{\partial \mathcal{L}}{\partial \lambda} = \sum\limits_{i=1}^{K} \pi_i - 1 = 0. \end{cases} \tag{C.39}$$

The solution to the above equation set can be written as:

$$\begin{cases} \pi_j = \dfrac{\sum_{i=1}^{N^t} g_{ij}^{(m)} + \alpha_j^{\mathbf{in}} - 1}{\lambda} \\ \rho_t = \dfrac{\sum_{i=1}^{N^t}\sum_{j=1}^{K} g_{ij}^{(m)} + \alpha_1^{\mathbf{out}} - 1}{N^t + \alpha_1^{\mathbf{out}} + \alpha_2^{\mathbf{out}} - 2} \\ \lambda = \sum_{i=1}^{N^t}\sum_{j=1}^{K} g_{ij}^{(m)} + \sum_{l=1}^{K}(\alpha_l^{\mathbf{in}} - 1). \end{cases} \tag{C.40}$$

Therefore optimal $\boldsymbol{\pi}, \rho_t$ for $Q(\boldsymbol{\pi}, \rho_t | \boldsymbol{\pi}^{(m)}, \rho_t^{(m)})$ without inequality constraints is given by:

$$\begin{aligned} \pi_j &= \frac{\sum_{i=1}^{N^t} g_{ij}^{(m)} + \alpha_j^{\mathbf{in}} - 1}{\sum_{i=1}^{N^t}\sum_{j=1}^{K} g_{ij}^{(m)} + \sum_{l=1}^{K}(\alpha_l^{\mathbf{in}} - 1)}, \\ \rho_t &= \frac{\sum_{i=1}^{N^t}\sum_{j=1}^{K} g_{ij}^{(m)} + \alpha_1^{\mathbf{out}} - 1}{N^t + \alpha_1^{\mathbf{out}} + \alpha_2^{\mathbf{out}} - 2} \end{aligned} \tag{C.41}$$

Proof that the solution satisfies inequality constraints Note that we have:

- $g_{ij}^{(m)}$ in Eq. (C.35) is non-negative
- $\tilde{c}_i > 0, i = 1, 2...K$ is non-negative
- $\alpha_i^{\mathbf{in}} - 1 > 0, i = 1, 2...K$ and $\alpha_1^{\mathbf{out}} - 1 > 0, \alpha_2^{\mathbf{out}} - 1 > 0$

Therefore we have $\boldsymbol{\pi}^{(t)} > 0 \Rightarrow \boldsymbol{\pi}^{(m+1)} > 0$. Because the optimization problem is convex, when $\pi_j^{(t)} > 0, j = 1, 2, ...K$, Eq. (C.41) gives the global optimal $\boldsymbol{\pi}^{(m+1)}, \rho_t^{(m+1)}$ for the optimization problem in Eq. (C.37):

$$\begin{cases} \pi_j^{(m+1)} = \dfrac{\sum_{i=1}^{N^t} g_{ij}^{(m)} + \alpha_j^{\mathbf{in}} - 1}{N - \sum_{i=1}^{N^t} g_{iK+1}^{(m)} + \sum_{l=1}^{K}(\alpha_l^{\mathbf{in}} - 1)} & \text{for all} \quad i \in \mathcal{Y} \\ \rho_t^{(m+1)} = \dfrac{N - \sum_{i=1}^{N^t} g_{iK+1}^{(m)} + \alpha_1^{\mathbf{out}} - 1}{N^t + \alpha_1^{\mathbf{out}} + \alpha_2^{\mathbf{out}} - 2}, \end{cases} \tag{C.42}$$

given the fact that $\sum_{i=1}^{N^t}\sum_{j=1}^{K} g_{ij}^{(m)} = N - \sum_{i=1}^{N^t} g_{iK+1}^{(m)}$.

□

C.1.6 Proof of Theorem 11 (See page 88)

Theorem 11. ***(Target ID/OOD ratio correction)*** *Under Assumption 1 and Assumption 5A (without 5B), for a classifier $h' : \mathcal{X} \to [0, 1]$ that satisfies Eq. (5.12), given source ID dataset $\mathcal{D}^s$, OOD dataset $\mathcal{D}^o$, target dataset $\mathcal{D}^t$, then for $\delta > 0$, with*

probability of at least $1-2\delta$ *we have:*

$$|\rho_t - \hat{\rho}_t^*| \leq \frac{1}{|\mu_1' - \mu_0'|}\sqrt{\frac{2\log 1/\delta}{\min(|\mathcal{D}^s|, |\mathcal{D}^o|, |\mathcal{D}^t|)}}, \tag{5.13}$$

where

$$\hat{\rho}_t^* = \frac{\hat{\rho}' - \hat{\mu}_0'}{\hat{\mu}_1' - \hat{\mu}_0'}, \quad \textit{and} \quad \hat{\rho}' := \frac{1}{|\mathcal{D}^t|}\sum_{x_i \in \mathcal{D}^t} h'(x_i), \tag{5.14}$$

with

$$\hat{\mu}_0' := \frac{1}{|\mathcal{D}^o|}\sum_{x \in \mathcal{D}^o} h'(x), \qquad \hat{\mu}_1' := \frac{1}{|\mathcal{D}^s|}\sum_{x \in \mathcal{D}^s} h'(x), \tag{5.15}$$

and

$$\mu_0' := \mathbb{E}_{X_s|B_s=0}[h'(x)], \qquad \mu_1' := \mathbb{E}_{X_s|B_s=1}[h'(x)]. \tag{5.16}$$

Proof. For target domain dataset $\mathcal{D}^t$, if we are given ID/OOD label: $\mathcal{D}^t = \mathcal{D}^{\mathbf{ti}} \cup \mathcal{D}^{\mathbf{to}}$, for a practical classifier $h'(x)$ we can write:

$$\begin{aligned}\rho' := \mathbb{E}_{X_t}[h'(x)] &= \mathbb{E}_{X_t|B_t=1}[h'(x)] \cdot p_t(b=1) + \mathbb{E}_{X_t|B_t=0}[h'(x)] \cdot p_t(b=0) \\ &= \rho_t \cdot \mathbb{E}_{X_t|B_t=1}[h'(x)] + (1-\rho_t) \cdot \mathbb{E}_{X_t|B_t=0}[h'(x)] \\ &= \mu_1^t \cdot \rho_t + \mu_0^t \cdot (1-\rho_t),\end{aligned} \tag{C.43}$$

where:

$$\mu_1^t := \mathbb{E}_{X_t|B_t=1}[h'(x)] \quad \text{and} \quad \mu_0^t := \mathbb{E}_{X_t|B_t=0}[h'(x)]. \tag{C.44}$$

Rearranging Eq. (C.43), we have that:

$$\rho_t = \frac{1}{\mu_1^t - \mu_0^t}\rho' - \frac{\mu_0^t}{\mu_1^t - \mu_0^t}, \tag{C.45}$$

where the equation holds when $\mu_1^t \neq \mu_0^t$:

$$\mu_1^t = \mathbb{E}_{X_t|B_t=1}[h'(x)] \neq \mathbb{E}_{X_t|B_t=0}[h'(x)] = \mu_0^t. \tag{C.46}$$

Option 1 Eq. (5.12): Under Assumption 1, the condition Eq. (5.12) holds implies:

$$\mathbb{E}_{X_s|Y_s=i}[h'(x)] = \mathbb{E}_{X_t|Y_t=j}[h'(x)] \quad \text{for all} \quad i,j \in \mathcal{Y}, \tag{C.47}$$

then according Eq. (5.1) we have:

$$\begin{aligned}\mu_1^t = \mathbb{E}_{X_t|B_t=1}[h'(x)] &= \sum_{i=1}^{K} \mathbb{E}_{X_t|Y_t=i}[p_t(y=i|b=1)\cdot h'(x)] \\ &= \sum_{i=1}^{K} \mathbb{E}_{X_t|Y_t=i}[\pi_i \cdot h'(x)] = \sum_{i=1}^{K} \pi_i \cdot \mathbb{E}_{X_t|Y_t=1}[h'(x)] \\ &= \mathbb{E}_{X_t|Y_t=1}[h'(x)] = \mathbb{E}_{X_s|Y_s=1}[h'(x)] \\ &= \sum_{i=1}^{K} c_i \mathbb{E}_{X_s|Y_s=1}[h'(x)] = \sum_{i=1}^{K} \mathbb{E}_{X_s|Y_s=i}[c_i \cdot h'(x)] \\ &= \sum_{i=1}^{K} \mathbb{E}_{X_s|Y_s=i}[p_s(y=i|b=1)\cdot h'(x)] \\ &= \mathbb{E}_{X_t|B_t=1}[h'(x)] = \mu_1'.\end{aligned} \tag{C.48}$$

Option 2 $\pi = c$ Condition Eq. (5.1) can actually be replace with $\pi = c$ with the results still holds:

$$\begin{aligned}\mu_1^t = \mathbb{E}_{X_t|B_t=1}[h'(x)] &= \sum_{i=1}^{K} \mathbb{E}_{X_t|Y_t=i}[p_t(y=i|b=1)\cdot h'(x)] \\ &= \sum_{i=1}^{K} \mathbb{E}_{X_t|Y_t=i}[\pi_i \cdot h'(x)] = \sum_{i=1}^{K} \mathbb{E}_{X_s|Y_s=i}[c_i \cdot h'(x)] \\ &= \sum_{i=1}^{K} \mathbb{E}_{X_s|Y_s=i}[p_s(y=i|b=1)\cdot h'(x)] \\ &= \mathbb{E}_{X_t|B_t=1}[h'(x)] = \mu_1',\end{aligned} \tag{C.49}$$

where $\mu_1' := \mathbb{E}_{X_s|B_s=1}[h'(x)]$.

For both options we have:

$$\begin{aligned}\mu_0^t = \mathbb{E}_{X_t|B_t=0}[h'(x)] &= \mathbb{E}_{X_t|Y_t=K+1}[p_t(y=K+1|b=0)\cdot h'(x)] \\ &= \mathbb{E}_{X_s|Y_s=K+1}[p_s(y=K+1|b=0)\cdot h'(x)] \\ &= \mathbb{E}_{X_s|B_s=0}[h'(x)] = \mu_0'.\end{aligned} \tag{C.50}$$

where $\mu_0' := \mathbb{E}_{X_s|B_s=0}[h'(x)]$ are defined in the same way as μ_1, μ_0 defined in Theorem 8 but substitute h as h'.

The expectations can be approximated by $\hat{\mu}_1', \hat{\mu}_1'$ with source domain ID dataset $\mathcal{D}^s$ and OOD dataset $\mathcal{D}^{\mathbf{o}}$ (Eq. (5.3)). Moreover, $\mathbb{E}_{X_t}[h(x)] = \rho$ can be estimated with $\hat{\rho}$ given target dataset $\mathcal{D}^t$:

$$\hat{\mu}_1' := \frac{1}{|\mathcal{D}^s|}\sum_{x_i\in\mathcal{D}^s} h'(x_i), \quad \hat{\mu}_0' := \frac{1}{|\mathcal{D}^{\mathbf{o}}|}\sum_{x_i\in\mathcal{D}^{\mathbf{o}}} h'(x_i) \text{ and } \hat{\rho}' := \frac{1}{|\mathcal{D}^t|}\sum_{x_i\in\mathcal{D}^t} h'(x_i). \tag{C.51}$$

Therefore as long as Eq. (5.12) holds, we can use $\mathcal{D}^s$ and $\mathcal{D}^{\mathbf{o}}$ to estimate ρ_t with Eq. (C.45):

$$\rho_t \approx \hat{\rho_t}^* := \frac{1}{\hat{\mu}'_1 - \hat{\mu}'_0}\hat{\rho}' - \frac{\hat{\mu}'_0}{\hat{\mu}'_1 - \hat{\mu}'_0}, \tag{C.52}$$

Note that since $h'(x) \in [0,1]$, the Hoeffding's inequality guarantees for some small $\epsilon > 0$:

$$\begin{aligned} p\left(|\mu'_0 - \hat{\mu}'_0| \geq \epsilon\right) &\leq 2e^{-2|\mathcal{D}^{\mathbf{so}}|\epsilon^2} \\ p\left(|\mu'_1 - \hat{\mu}'_1| \geq \epsilon\right) &\leq 2e^{-2|\mathcal{D}^{\mathbf{si}}|\epsilon^2} \\ p\left(|\rho' - \hat{\rho}'| \geq \epsilon\right) &\leq 2e^{-2|\mathcal{D}^{t}|\epsilon^2}, \end{aligned} \tag{C.53}$$

Therefore with high probability of at least $1 - 2e^{-2\min(|\mathcal{D}^{\mathbf{si}}|,|\mathcal{D}^{\mathbf{so}}|,|\mathcal{D}^{t}|)\epsilon^2}$ we have:

$$\begin{cases} \rho_t - \hat{\rho}^*_t \leq \dfrac{|\rho' - \mu'_0|}{|\mu'_1 - \mu'_0|} - \dfrac{|\rho' - \epsilon - \mu'_0 - \epsilon|}{|\mu'_1 + \epsilon - \mu'_0 - \epsilon|} = \dfrac{2\epsilon}{|\mu'_1 - \mu'_0|} \\ \rho_t - \hat{\rho}^*_t \geq \dfrac{-|\rho' - \mu'_0|}{|\mu'_1 - \mu'_0|} - \dfrac{-|\rho' + \epsilon - \mu'_0 + \epsilon|}{|\mu'_1 - \epsilon - \mu'_0 + \epsilon|} = \dfrac{-2\epsilon}{|\mu'_1 - \mu'_0|}, \end{cases} \tag{C.54}$$

which is equivalent to:

$$|\rho_t - \hat{\rho}^*_t| \leq \frac{2\epsilon}{|\mu'_1 - \mu'_0|}. \tag{C.55}$$

Letting $\delta := e^{-2\min(|\mathcal{D}^{\mathbf{si}}|,|\mathcal{D}^{\mathbf{so}}|,|\mathcal{D}^{t}|)\epsilon^2}$, rearrange the equations and we get the result.

□

C.1.7 Further Discussion on ρ_t correction model

This section further discuss the ρ_t correction model Eq. (5.17) proposed in §5.3.4 in our main paper. The model adjust ρ_t^{MLE} and ρ_t^{MAP} obtained in Alg. 7 with Eq. (5.17), which based on Theorem 11.

We will show that for a special case, the MLE ρ_t^{MLE} defined in MLE objective Eq. (5.9) will have close form solution, which is simply averaging the response of $h(x)$ on target dataset $\mathcal{D}^t$:

Lemma 14. *Under Assumption 1,5, if $\boldsymbol{\pi} = \boldsymbol{c}$ and $h : \mathcal{X} \to \{0,1\}$, then the ρ_t^{MLE} defined in Eq. (5.9) can be obtained given target dataset $\mathcal{D}^t$ via:*

$$\rho_t^{MLE} = \frac{1}{N^t}\sum_{i=1}^{N^t} h(x_i). \tag{C.56}$$

Proof. When Assumption 5B is satisfied, given the information available, substituting:

$$p_s(b = 1|x) = h(x) \in \{0,1\} \quad \text{and} \quad \boldsymbol{\pi} = \boldsymbol{c} \tag{C.57}$$

into the NLL in Eq. (5.6) and we have:

$$\begin{aligned}
-\log L(\rho_t; \mathcal{D}^t) &= -\sum_{i=1}^{N^t} \log \left(\frac{\rho_t}{\rho_s} h(x_i) \cdot \sum_{j=1}^{K} \frac{\pi_j}{c_j} f(x_i)_j + \frac{1-\rho_t}{1-\rho_s} \cdot (1 - h(x_i)) \right) \\
&\quad + Const \\
&= -\sum_{i=1}^{N^t} \log \left(\frac{\rho_t}{\rho_s} h(x_i) \cdot 1 + \frac{1-\rho_t}{1-\rho_s} \cdot (1 - h(x_i)) \right) + Const \\
&= -\sum_{i=1}^{N^t} \mathbb{I}_1(h(x_i)) \cdot \log \left(\frac{\rho_t}{\rho_s} \right) - \sum_{i=1}^{N^t} (1 - \mathbb{I}_1(h(x_i))) \\
&\quad \times \log \left(\frac{1-\rho_t}{1-\rho_s} \right) + Const
\end{aligned} \tag{C.58}$$

Let the derivative w.r.t. ρ_t equals 0 and we have:

$$\begin{aligned}
\frac{d\left(-\log L(\rho_t; \mathcal{D}^t)\right)}{d\rho_t} &= -\sum_{i=1}^{N^t} \mathbb{I}_1(h(x_i)) \cdot \frac{\rho_s}{\rho_t} \cdot \frac{1}{\rho_s} - \sum_{i=1}^{N^t} (1 - \mathbb{I}_1(h(x_i))) \\
&\quad \times \frac{1-\rho_s}{1-\rho_t} \cdot \frac{-1}{1-\rho_s} \\
&= -\sum_{i=1}^{N^t} \mathbb{I}_1(h(x_i)) \cdot \frac{1}{\rho_t} + \sum_{i=1}^{N^t} (1 - \mathbb{I}_1(h(x_i))) \cdot \frac{1}{1-\rho_t} \\
&= 0
\end{aligned} \tag{C.59}$$

Solve the above equation for ρ_t and we get:

$$\rho_t = \frac{1}{N^t} \sum_{i=1}^{N^t} h(x_i), \tag{C.60}$$

which is the close form solution to the MLE objective Eq. (5.6) under special setting of no ID label shift ($\boldsymbol{\pi} = \boldsymbol{c}$) and discrete ID/OOD classifier ($h : \mathcal{X} \to \{0, 1\}$). □

As shown in Lemma 14, when $\boldsymbol{\pi} = \boldsymbol{c}$ and $h : \mathcal{X} \to \{0, 1\}$, ρ_t^{MLE} can be obtained by averaging $h(x) = p_s(b = 1|x)$ over the target dataset $\mathcal{D}^t$ based on Assumption 5B, *i.e.* $h(x) \neq p_s(b = 1|x)$ when the assumption is not satisfied, according the proof of Theorem 11, the condition $\boldsymbol{\pi} = \boldsymbol{c}$ enable us to use:

$$\hat{\rho}_t^* = \frac{\hat{\rho} - \hat{\mu}_0'}{\hat{\mu}_1' - \hat{\mu}_0'}, \quad \text{and} \quad \hat{\rho}' := \frac{1}{|\mathcal{D}^t|} \sum_{x_i \in \mathcal{D}^t} h(x_i), \tag{C.61}$$

to obtain the estimate of ground truth ρ_t.

Dataset	Model	Setup	optimizer	lr	weight decay	epoch
CIFAR10	ResNet18	Train from Scratch	SGD	0.1	$5e^{-4}$	100
CIFAR100	ResNet18	Train from Scratch	SGD	0.1	$5e^{-4}$	100
ImageNet-200	ResNet18	Train from Scratch	SGD	0.1	$5e^{-4}$	90

Table C.1: Soure domain ID classifier f setup used in our model.

C.2 Detailed Experimental Setups

ID Classifier Details : Our code of training ID classifier and constructing OOD classifier is mainly based on the open source project OpenOOD (Yang, Wang et al., 2022; Zhang, Yang et al., 2023) on OOD detection. The project is publicly available in https://github.com/Jingkang50/OpenOOD.

We follows the basic setup in OpenOOD to train ID classifier f, where we train a ResNet18 model for CIFAR10/100 and ImageNet-200 datasets. Each model is trained 3 times with different random seeds.

OOD Classifier Details : We use the implementation provided in OpenOOD project to construct the OOD detection binary classifiers h proposed by OpenMax (Bendale and Boult, 2016), Ash (Djurisic et al., 2022), MLS (Hendrycks, Basart et al., 2019a), ReAct (Sun, Guo et al., 2021) and KNN (Sun, Ming et al., 2022). All the OOD detection models are pos-hoc inference models based on the ID classifier f. The detailed hyper-parameter setups of each OOD detector are listed in Tab. C.3.

Model Name	Source Code	Date of Retrieval
OpenOOD Yang, Wang et al. (2022)	https://github.com/Jingkang50/OpenOOD	May 2024
OpenMax Bendale and Boult (2016)	https://github.com/Jingkang50/OpenOOD	May 2024
KNN Sun, Ming et al. (2022)	https://github.com/deeplearning-wisc/knn-ood	May 2024
MLS Hendrycks, Basart et al. (2019a)	https://github.com/Jingkang50/OpenOOD	May 2004
Ash Djurisic et al. (2022)	https://github.com/andrijazz/ash	May 2024
ReAct Sun, Guo et al. (2021)	https://github.com/deeplearning-wisc/react	May 2024

Table C.2: Source code details of reproduced OOD detection models. The code for OpenMax, KNN, MLS, Ash and ReAct have been collected in the OpenOOD project and can be directly tested within the project.

Output Re-scaling: Existing OOD classifiers focus more on ID/OOD separation and hence usually output a real valued scalar instead of $[0,1]$ confidence. For example, MLS (Hendrycks, Basart et al., 2019a) model actually outputs the max logit of the ID classifier's prediction. To use these OOD classifiers in our OSLS estimation model, we need to re-scale the output of these OOD classifier in the binary range $[0,1]$.

In this work, we re-scale a OOD classifier $h' : \mathcal{X} \to \mathbb{R}$ to a binary classifier $h : \mathcal{X} \to [0,1]$ with two approaches: **logistic regression** and **thresholding**. The logistic regression model $h_0 : \mathbb{R}_+ \to [0,1]$ is trained based on the source domain ID dataset $\mathcal{D}^s$ and reference OOD dataset $\mathcal{D}^{\mathbf{o}}$ (see Fig. 5.3). On the other hand, the thresholding approach obtain the threshold by computing the median values of the output of OOD classifier h' given ID dataset $\mathcal{D}^s$ and OOD dataset $\mathcal{D}^{\mathbf{o}}$. The threshold is picked as the average of the two median values. Details or the two re-scaling models are described in Tab. C.4.

OOD classifier	hyper-parameters
OpenMax	Weibull fitting: alpha=3, threshold=0.9, tail=20; coreset_sampling_ratio=0.01;
KNN	# of nearest neighbor $K = 50$
MLS	-
Ash	parameter search on percentile=[65, 70, 75, 80, 85, 90, 95]
ReAct	parameter search on percentile=[85, 90, 95, 99]

Table C.3: Detailed hyper-parameter setups of the OOD detectors used in our work. All the hyper-parameter setup are following the default setups provided by the OpenOOD project (Yang, Wang et al., 2022).

Dataset	Re-scaling Model	Model Setup
CIFAR10/100	Logistic Regression	epoch 100; optimizer: SGD; batch_size: 512; lr 0.05; lr_scheduler: Cosine; loss: BCE; $h(x) = 1/(1+e^{-w\cdot h'(x)+b})$
ImageNet-200	Thresholding	$h(x) = \begin{cases} 1, & h'(x) > (\text{median}(h'(\mathcal{D}^s)) + \text{median}(h'(\mathcal{D}^{\mathbf{o}})))/2 \\ 0, & \text{Otherwise} \end{cases}$.

Table C.4: Re-scaling model setup that normalize the output of a OOD classifier into the continuous range $[0, 1]$.

Although thresholding approach only output $\{0, 1\}$ instead of a continuous confidence, it's suitable for our ρ_t correction model (Sec. 5.3.4) because the linear correction approach have theoretical guarantees when the ID/OOD classifier $h(x)$ output binary values (Theorem 11). Moreover, OOD detectors on large-scale datasets are more likely to violate Assumption 5, thus ρ_t correction model might become more necessary.

OOD reference Dataset details : As discussed in the main paper, the reference OOD dataset $\mathcal{D}^{\mathbf{o}}$ is generated by linear combination of Gaussian noise and ground truth samples in source domain ID dataset $\mathcal{D}^s$. The hyper-parameters used γ, T used in the OOD dataset generation process and $\hat{\mu}_0$ rescaling are: CIFAR10: $\gamma = 0.2, T = 2$, CIFAR100: $\gamma = 0.1, T = 2$, ImageNet-200: $\gamma = 0.2, T = 2$.

Datasets Details We use the standard CIFAR10/100 and ImageNet-200 datasets as ID datasets, with te detailed information provided in Tab. C.5.

We use the OOD datasets setup provided in the OpenOOD project, where validation sets of CIFAR10 and CIFAR100 are used as OOD datasets, with each contains 9000 samples. For the other OOD datatsets, TinyImageNet has 7793 samples, MNIST has 70000 samples, SVHN has 26032 samples, Texture has 5640 samples, Places has 35195 samples, SSB has 49000 samples, NINCO has 5879 sampels, iNaturalist has 10000 samples, OpenImage-O has 15869 samples. Many of these OOD datasets are actually subsampled from the original datasets to avoid overlapping in classes.

Closed Set Label Shift Estimation Model details We test the closed set label shift estimation models MAPLS (Chapter 3), MLLS (Saerens et al., 2002), BBSE (Lipton et al., 2018) and RLLS (Azizzadenesheli et al., 2018) based on the official implementation of MAPLS provided in `https://github.com/ChangkunYe/MAPLS` retrieved at Jun 2024. These models are directed used to test on the open set label shift dataset without any adjustment on hyper-parameters or other setups.

Dataset	Train # samples	Val # samples	Test # samples	# of Classes
CIFAR10	50k	9000	1000	10
CIFAR100	50k	9000	1000	100
ImageNet-200	260k	1000	9000	200

Table C.5: Detailed information of ID datasets.

EM algorithm We use the same EM algorithm running procedure as MAPLS (Ye et al., 2024) proposed in the closed set label shift problem. Specifically, the procedure is as follows: 1) Initialize the target label distribution to be the same as source label distribution, *i.e.* $\boldsymbol{\pi}^{(0)} = \hat{\boldsymbol{c}}$ and $\rho_t^{(0)} = \hat{\rho}_s$, 2) Run EM algorithm 7 for 100 epoch to ensure convergence and 3) Output $\boldsymbol{\pi}^{(101)}$ and $\boldsymbol{\pi}^{(101)}$.

For MAP estimate, we use the Adaptive Prior Learning (APL) model proposed by MAPLS (Ye et al., 2024) to determine parameter $\boldsymbol{\alpha}^{\mathbf{in}} \in \mathbb{R}^K_{>1}$ in the Dirichlet prior for ID classes and use no Bernoulli prior ($\alpha_1^{\mathbf{out}}, \alpha_2^{\mathbf{out}} = \mathbf{1}$).

Evaluation Metrics **Label Shift Estimation Error:** The label shift estimation error $(w - \hat{w})^2 / K$, is widely used in evaluating the closed set label shift estimation models (Alexandari et al., 2020; Lipton et al., 2018), where $w = \boldsymbol{\pi} / \boldsymbol{c}$ is the ground truth target over source label distribution ratio and $\hat{w}$ is the corresponding estimator. This work evaluate the open set label shift estimation model in terms of label shift estimation error on ID classes, where source ID label distribution $p_s(y = \cdot | b = 1; \boldsymbol{c}) = \boldsymbol{c}$ have ground truth i.i.d. samples available in source labeled dataset $\mathcal{D}^s$.

Top1 Accuracy: The Top1 Accuracy of each label shift estimation model is obtained by first estimating the label shift with corresponding model, then correct label shift on target domain for classifier $\tilde{f}$, with offline label shift correction method defined in Eq. Eq. (2.34). We also report Top1 Accuracy on baseline classifier without any label shift correction.

Appendix D

Appendix for Chapter 6

D.1 Detailed Experimental Setups

In Chapter 6, we reproduced the performance of several SOTA models using their published code on Proposed Split V2.0. They are E-PGN (Yu et al., 2020), Li, Min et al. (2019) and DVBE (Min et al., 2020). The detailed information for these published models is available in Table D.1 below.

We reproduce the results by precisely following the instructions provided by the authors of each model, with the exception that we use a different dataset split (Proposed Splits V2.0: https://drive.google.com/file/d/1p9gtkuHCCCyjkyezSarCw-1siCSXUykH/view?usp=sharing), updated by Xian, Lampert et al. (2019). We fine-tune hyperparameters for "Proposed Split V2.0" by parameter search around values recommended for "Proposed Split" in each official code. We note that some models like E-PGN (Yu et al., 2020) are sensitive to random seeds and difficult to fine-tune. Hence, despite our best efforts, performance may be sub-optimal.

The hyperparameters and corresponding values used to reproduce performance for each model Li, Min et al. (2019), EPGN (Yu et al., 2020), DVBE (Min et al., 2020) are listed in Table D.2, D.3 and D.4 respectively.

D.2 More Ablation Study

D.2.1 Ablation Study on model structure

We report the Area Under the Seen and Unseen Curve (AUSUC) of our model, along with some alternative model structures. These models include the Kernel Ridge Regression (KRR) model, the KRR model performed on a latent space that was trained with our proposed $\mathcal{L}_{BT}$ triplet loss, the Gaussian Process (GP) model and a GP model performed on a latent space trained using the original triplet loss $\mathcal{L}_{T}$.

Model	Conference	Code Link	Time of Retrieval
Li, Min et al. (2019)	CVPR 19	https://github.com/kailigo/cvcZSL	Dec 2020
E-PGN (Yu et al., 2020)	CVPR 20	https://github.com/yunlongyu/EPGN	Dec 2020
DVBE (Min et al., 2020)	CVPR 20	https://github.com/mboboGO/DVBE	Dec 2020

Table D.1: Official, published, code links and time of code retrieval for each reproduced SOTA model in Chapter 6.

Params	ways	shots	lr	opt_decay	step_size	model_file
AWA1	16	4	1e-5	1e-4	500	lr5_opt4_ss500_w16_s4.pt

Table D.2: Hyperparameters used for reproducing Li, Min et al. (2019) on AWA1 dataset with Proposed Split V2.0. The name of each hyperparameter matches with the published code.

Params	mid_dim	hid_dim	lr	epoch	episode	inner_loop	batch_size	dropout	manualSeed
CUB	1600	1800	5e-5	15	100	10	32	True	4196
AWA1	1200	1800	5e-5	30	50	100	100	True	4198
AWA2	1800	1800	2e-4	30	50	30	64	True	4198

Table D.3: Hyperparameters used for reproducing EPGN (Yu et al., 2020) on CUB, AWA1 and AWA2 datasets with Proposed Split V2.0. The name of each hyperparameter matches with the published code.

As can be seen from Figure D.1, GP-based models consistently perform better than KRR-based models. Also, our proposed triplet loss $\mathcal{L}_{BT}$ can generally improve the performance of the KRR model as well as the GP model. Our proposed model has improvements in both the seen accuracy and the unseen accuracy compared with other alternative models.

D.2.2 Normalizing Unbounded Feature Space

In our ZSL model, the feature vector of each image is extracted by a pre-trained ResNet101 model, proposed by Xian, Lampert et al. (2019). The normalised histogram of feature values is shown in Figure D.2 and some statistical metrics are given in Table D.5. As argued by Le Cacheux et al. (2019b), unbounded feature values may prevent Neural Network models from learning using triplet loss.

One way to sidestep this problem is to simply bound the feature space. As shown in Table D.5, we found that 99.90% of the feature values are below a threshold of about 7 for each dataset. We thus propose to bound the feature space by preprocessing feature vectors for each dataset by clipping values above 7 before normalising to the range $[0, 1]$. This is a simpler approach compared with the partial normalisation approach proposed by Le Cacheux et al. (2019b).

D.2.3 Ablation Study on hyperparameters

We report more detailed results for the methods reported in the ablation study section in the main paper. In Table D.6, we report the performance of our model influenced by different clipping values used in data preprocessing. As discussed in the previous section, the objective of clipping is to create a bounded feature space such that our

Params	batch_size	lr1	lr2	momentum	epochs	epoch_decay	sigma	weight_decay	workers	seed
CUB	128	0.1	0.001	0.9	90	30	0.5	0.0001	3	5181
AWA2	128	0.1	0.001	0.9	90	30	0.5	0.0001	3	142
APY	128	0.1	0.001	0.9	90	30	0.5	0.0001	3	119

Table D.4: Hyperparameters used for reproducing DVBE (Min et al., 2020) on CUB, AWA2 and APY datasets with Proposed Split V2.0. The name of each hyperparameter matches with the published code.

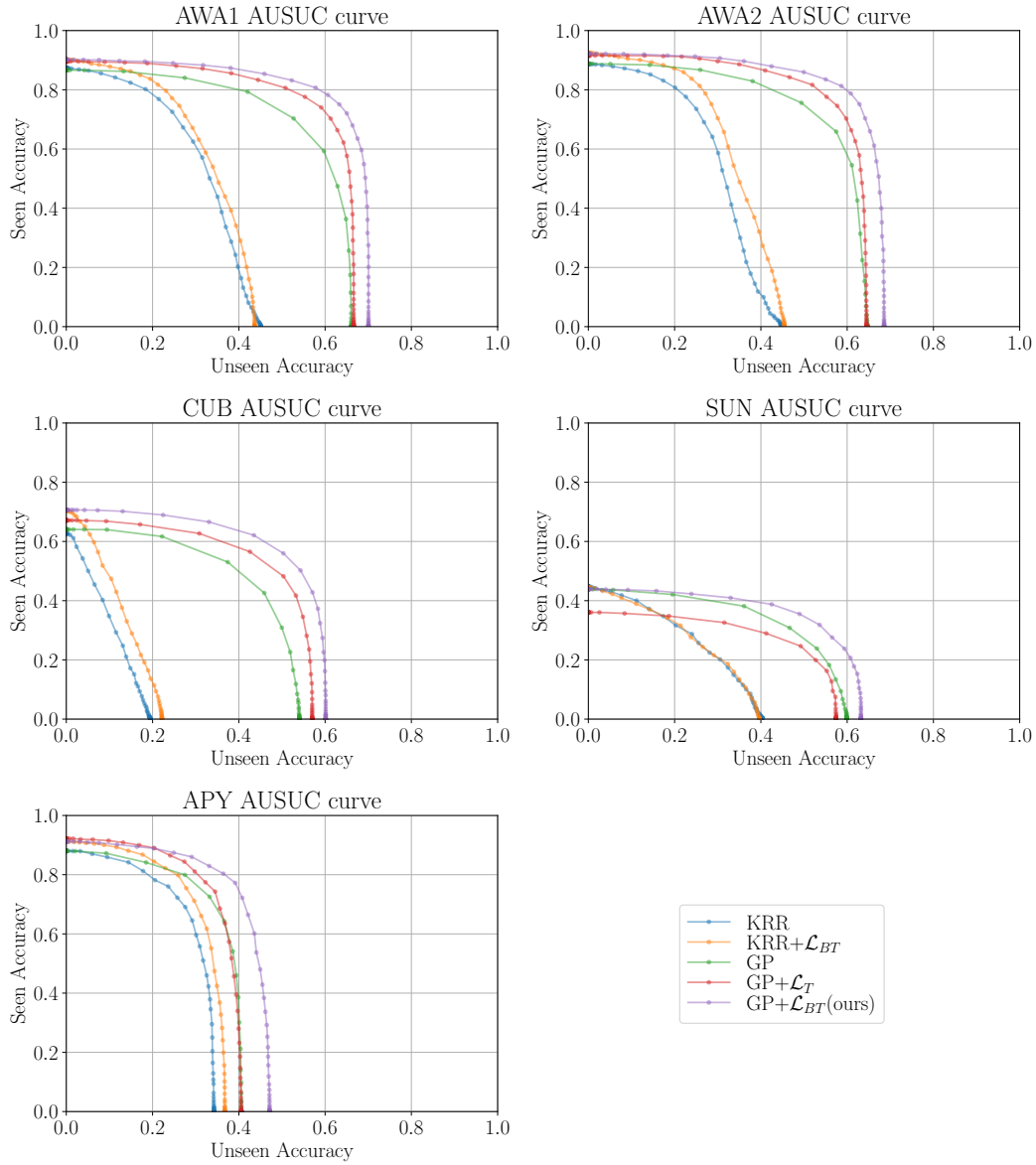

Figure D.1: Area Under Seen and Unseen Curve (AUSUC) for different model structures in the ablation study section in the main paper. Our model is consistently better than alternative structures. Moreover, KRR and GP models achieve better performances when the latent space is trained using our proposed $\mathcal{L}_{BT}$ loss compared with the original feature space

Dataset	CUB	SUN	AWA2	AWA1	APY
Average feature value	0.3293	0.4413	0.4049	0.4244	0.4459
Maximum feature value	32.95	44.83	61.00	47.21	46.55
99.90% feature value lies in range	$[0.00, 6.25]$	$[0.00, 7.81]$	$[0.00, 7.09]$	$[0.00, 7.00]$	$[0.00, 7.74]$

Table D.5: Analysis of feature vector values. Every dataset has maximal values that are too far away from the average, 99.90% of the values lie approximately in the range $[0, 7]$ for each dataset. Thus, we preprocess feature vectors for each dataset by clipping by seven and normalise to the range $[0, 1]$

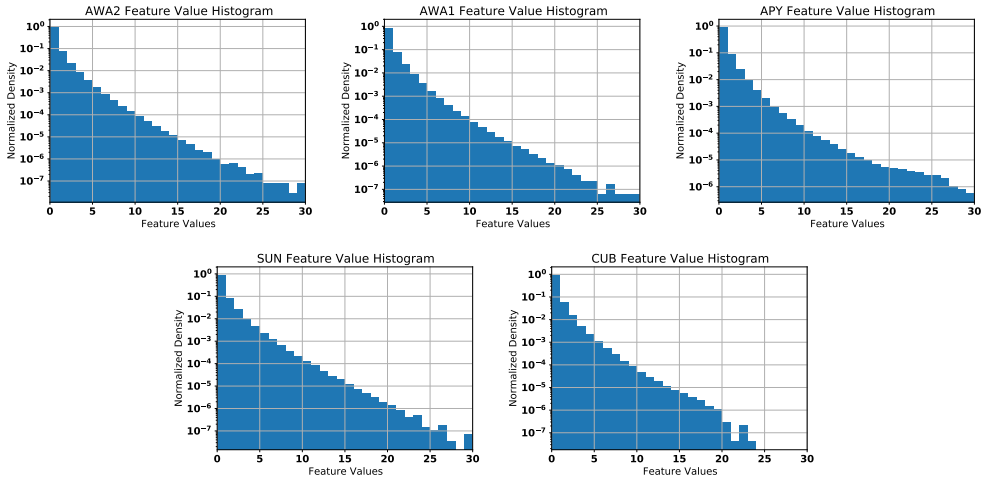

Figure D.2: Normalized histogram of feature vector values in each ZSL dataset. The probability density of feature values drops drastically as the feature value increases. Therefore, bounding the feature space by clipping the tail of the density requires modifying only a small amount of the values of the data.

Neural Network model can efficiently be trained using triplet loss. As shown in the table, the performance of our model is better when using feature clipping than without feature clipping.

In Table D.7, we report the performance of our model with different δ values in the triplet loss. In our model, δ is determined by an empirical grid search, with a coarse grid search in range $(0, 100]$, followed by a fine grid search in range $(0.25, 10]$. The hyperparameter δ serves as a threshold in the triplet loss. A triplet $\{x_1^i, x_2^i, x_1^j\}$ is a trivial triplet if the inter-class pair distance exceeds the intra-class pair distance by a given margin δ, *i.e.* $(x_1^i - x_1^j)^2 - (x_1^i - x_2^i)^2 > \delta$.

According to the triplet loss equation, trivial triplets will not influence backpropagation gradients of the Neural Network. Small values for δ may result in only a few non-trivial triplets, thus lowering the performance, while large δ may add unnecessary computational costs when training the model.

As shown in Table D.7, our model maintains a good performance with $\delta \geq 3$ on all ZSL datasets. The performance peaks at $\delta \approx 4$ and decreases slightly for larger δ applied in the triplet loss $\mathcal{L}_{BT}$.

D.3 Performance on the Incorrect “Proposed Split"

To ensure a fair comparison, in Table D.8, we also compare our model's performance on the original “Proposed Split" with results reported by previous SOTA papers, including f-VAEGAN-D2 (Xian, Sharma et al., 2019), RELATION NET (Sung et al., 2018), DAZLE (Huynh and Elhamifar, 2020), Li, Min et al. (2019), E-PGN (Yu et al., 2020), OCD (Keshari et al., 2020), DVBE (Min et al., 2020), TF-VAEGAN (Narayan et al., 2020), IZF (Shen et al., 2020), AGZSL (Chou, Lin et al., 2020), IPN (Liu, Zhou et al., 2020) and CE-GZSL (Han, Fu, Chen et al., 2021). We have not listed SOTA models that only report ImageNet performance like DGP (Kampffmeyer

Clip Value	CUB				SUN				AWA2				AWA1				APY			
	ZSL	GZSL			ZSL	GZSL			ZSL	GZSL			ZSL	GZSL			ZSL	GZSL		
	A_T	A_U	A_S	H	A_T	A_U	A_S	H	A_T	A_U	A_S	H	A_T	A_U	A_S	H	A_T	A_U	A_S	H
3	**61.2**	**52.0**	57.2	**54.5**	59.7	48.9	33.1	39.5	67.8	**63.5**	75.6	**69.0**	67.4	62.6	73.1	67.5	40.2	32.9	**72.7**	45.3
4	60.0	49.2	**58.3**	53.4	59.8	48.1	33.8	39.7	67.5	62.6	76.5	68.8	69.1	65.3	71.1	68.1	44.4	38.1	68.4	48.9
5	59.5	49.1	57.3	52.9	61.4	49.3	33.6	40.0	67.8	62.1	**77.2**	68.8	69.7	**66.0**	70.6	68.2	45.5	40.0	67.0	50.1
6	59.9	49.8	56.6	53.0	62.6	49.4	**35.2**	**41.1**	68.5	63.1	75.0	68.5	70.2	65.3	71.5	68.3	44.9	38.4	71.5	50.0
7	60.1	50.3	56.0	53.0	**63.2**	50.4	34.8	**41.1**	**68.6**	62.2	76.6	68.7	70.0	64.5	**73.3**	**68.6**	**47.1**	**42.8**	64.3	51.4
8	59.7	50.3	55.6	52.8	63.1	50.0	34.5	40.8	68.3	62.2	74.8	67.9	70.0	65.1	70.5	67.7	46.9	41.6	68.4	51.7
9	60.0	48.6	57.8	52.8	62.8	50.0	34.5	40.8	68.0	61.2	75.3	67.5	69.9	63.8	72.6	67.9	46.1	42.4	62.3	50.5
10	60.2	50.7	55.7	53.1	62.6	51.9	31.9	39.5	68.3	60.5	76.2	67.4	**70.7**	63.8	73.1	68.2	47.0	42.2	69.7	**52.6**
None	60.0	48.6	57.5	52.7	62.0	47.1	34.3	39.7	67.0	60.1	74.4	66.5	69.5	63.0	71.5	67.0	44.8	41.0	65.1	50.3

Table D.6: Ablation Study with Clip number selected during feature preprocessing, with all other parts of the model fixed. Our model has a better performance with feature clipping than without feature clipping in data preprocessing. The performance of our model is robust when varying the clipping value around the proposed threshold 7. Moreover, different clip values have only a slight influence on our model's performance.

δ Value	CUB				SUN				AWA2				AWA1				APY			
	ZSL	GZSL			ZSL	GZSL			ZSL	GZSL			ZSL	GZSL			ZSL	GZSL		
	A_T	A_U	A_S	H	A_T	A_U	A_S	H	A_T	A_U	A_S	H	A_T	A_U	A_S	H	A_T	A_U	A_S	H
0.25	58.2	**57.1**	33.2	42.0	60.3	**56.2**	22.6	32.3	66.2	**65.0**	48.0	55.2	69.3	68.3	49.5	57.4	40.3	38.5	37.9	38.2
0.5	59.1	52.6	50.7	51.6	60.5	56.8	21.9	31.6	67.7	64.7	62.6	63.6	69.2	66.7	60.4	63.4	42.1	39.0	53.8	45.2
1	59.3	53.9	49.9	51.8	61.3	54.9	28.2	37.3	67.2	61.2	74.5	67.2	69.2	63.4	71.9	67.4	44.7	40.2	61.3	48.5
2	59.8	51.8	53.8	52.8	62.3	49.9	34.5	40.8	67.9	62.5	73.6	67.6	70.0	65.4	70.3	67.8	46.6	41.2	66.4	50.8
3	59.9	50.7	55.1	52.8	62.6	50.6	34.6	41.1	68.5	62.3	75.4	68.2	**70.2**	64.7	72.5	68.4	45.3	40.6	65.8	50.2
4	**60.1**	50.3	56.0	**53.0**	63.2	50.4	34.8	41.1	68.6	62.2	76.6	**68.7**	70.0	64.5	**73.3**	**68.6**	**47.1**	**42.8**	64.3	**51.4**
5	59.9	49.6	56.2	52.7	63.2	50.3	**35.0**	**41.3**	68.9	61.9	76.6	68.5	69.5	64.0	72.6	68.0	45.9	41.9	62.6	50.2
6	59.5	49.2	56.2	52.5	63.4	51.6	34.0	41.0	**69.2**	62.0	76.9	68.6	69.8	**64.8**	71.8	68.1	46.3	40.2	**71.4**	**51.4**
7	59.5	48.8	**56.5**	52.4	**63.5**	51.1	34.5	41.2	68.7	62.0	76.6	68.5	70.1	**64.8**	72.3	68.3	44.5	40.2	66.2	50.0
8	59.4	48.5	**56.5**	52.2	63.2	50.6	34.5	41.1	69.0	62.0	76.5	68.5	69.9	64.3	73.0	68.3	45.1	39.4	70.8	50.6

Table D.7: Ablation Study with a threshold δ in the class balanced triplet loss, with all the other parts of the model fixed. As long as $\delta \geq 3$, our model has relatively stable performance.

et al., 2019) and HVE (Liu, Chen et al., 2020), or only report transductive ZSL results like SDGN (Wu, Zhang et al., 2020).

As can be seen from Table D.8, on "Proposed Split", our model has reached SOTA performance on SUN and APY datasets. By comparing the results shown in Table II in the main paper and Table D.8, it can be seen that all previous reproduced works have a performance decrease after switching from "Proposed Split" to the correct "Proposed Split V2.0". On the contrary, although our model also reports a performance decrease on fine-grained datasets CUB and SUN, it maintains relatively stable performance on coarse-grained datasets AWA1, AWA2 and APY. This may be because our model has a simple structure and is less prone to overfitting.

Methods	CUB				SUN				AWA2				AWA1				APY			
	ZSL	GZSL			ZSL	GZSL			ZSL	GZSL			ZSL	GZSL			ZSL	GZSL		
	A_T	A_U	A_S	H	A_T	A_U	A_S	H	A_T	A_U	A_S	H	A_T	A_U	A_S	H	A_T	A_U	A_S	H
SYNC	55.6	11.5	70.9	19.8	56.3	7.9	43.3	13.4	46.6	10.0	90.5	18.0	54.0	8.9	87.3	16.2	23.9	7.4	66.3	13.3
GFZSL	49.3	0.0	45.7	0.0	60.6	0.0	39.6	0.0	63.8	2.5	80.1	4.8	68.3	1.8	80.3	3.5	38.4	0.0	83.3	0.0
ALE	54.9	23.7	62.8	34.4	58.1	21.8	33.1	26.3	62.5	14.0	81.8	23.9	59.9	16.8	76.1	27.5	39.7	4.7	73.6	8.7
DEVISE	52.0	23.8	53.0	32.8	56.5	16.9	27.4	20.9	59.7	17.1	74.7	27.8	54.2	13.4	68.7	22.4	39.8	4.9	76.9	9.2
GDAN	-	39.3	66.7	49.5	-	38.1	**89.9**	53.4	-	32.1	67.5	43.5	-	-	-	-	-	30.4	75.0	43.4
CADA-VAE	-	51.6	53.5	52.4	-	47.2	35.7	40.6	-	55.8	75.0	63.9	-	57.3	72.8	64.1	-	-	-	-
TF-VAEGAN	64.9	52.8	64.7	58.1	66.0	45.6	40.7	43.0	-	-	-	-	72.2	59.8	75.1	66.6	-	-	-	-
f-VAEGAN-D2	61.0	48.4	60.1	53.6	65.6	50.1	37.8	43.1	-	-	-	-	71.1	57.6	70.6	63.5	-	-	-	-
RELATION NET	55.6	38.1	61.1	47.0	-	-	-	-	64.2	30.0	**93.4**	45.3	68.2	31.4	**91.3**	46.7	-	-	-	-
DAZLE	-	59.6	56.7	58.1	-	24.3	52.3	33.2	-	60.3	75.7	67.1	-	-	-	-	-	-	-	-
Li, Min et al. (2019)	54.4	47.4	47.6	47.5	60.8	42.6	36.6	39.4	71.1	56.4	81.4	66.7	70.9	62.7	77.0	69.1	38.0	26.5	74.0	39.0
E-PGN	**72.4**	52.0	61.1	56.2	-	-	-	-	73.4	52.6	83.5	64.6	**74.4**	62.1	83.4	**71.2**	-	-	-	-
DVBE	-	53.2	60.2	56.5	-	45.0	37.2	40.7	-	**63.6**	70.8	67.0	-	-	-	-	-	32.6	58.3	41.8
OCD	-	44.8	59.9	51.3	-	44.8	42.9	43.8	-	59.5	73.4	65.7	-	-	-	-	-	-	-	-
IZF-Softmax	67.1	52.7	68.0	59.4	**68.4**	52.7	57.0	54.8	**74.5**	60.6	77.5	68.0	74.3	61.3	80.5	69.6	44.9	42.3	60.5	49.8
AGZSL	57.2	41.4	49.7	45.2	63.3	29.9	40.2	34.3	73.8	65.1	78.9	71.3	-	-	-	-	41.0	35.1	65.5	45.7
IPN	-	60.2	73.8	66.3	-	-	-	-	-	67.5	79.2	**72.9**	-	-	-	-	-	37.2	66.0	47.6
CE-GZSL	-	63.9	66.8	65.3	-	48.8	38.6	43.1	-	63.1	78.6	70.0	-	**65.3**	73.4	69.1	-	-	-	-
Ours	61.0	51.1	71.0	59.4	64.3	**53.6**	61.6	**57.3**	67.9	61.1	78.3	68.6	71.2	64.5	76.1	**69.8**	**48.4**	**42.6**	**74.5**	**54.2**

Table D.8: Zero-Shot Learning Top-1 per-class Accuracy on incorrect “Proposed Split". The results of each model are reported by original papers. Although our model is less prone to overfitting, we still outperform previous papers on SUN and APY datasets. Some works are not included in Table II due to the unavailable published official code.