NATURAL ROBUSTNESS OF MACHINE LEARNING IN THE OPEN WORLD

HONGXIN WEI

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2023

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

06/01/2023

Date

HONGXIN WEI

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

06/01/2023

Date

Prof. Bo An

Authorship Attribution Statement

This thesis contains material from 5 papers published in the following peer-reviewed journal(s) / from papers accepted at conferences in which I am listed as an author.

Chapter 3 is published as Hongxin Wei, Lei Feng, Xiangyu Chen, Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13726-13735. 2020. The contributions of the co-authors are as follows:

- I proposed the improved solutions, conducted the experiments, and wrote the main manuscript.
- Dr. Lei Feng provided the initial project direction and revised the manuscript drafts.
- Xiangyu Chen assisted in conducting the experiments.
- Prof. Bo An provided insightful comments and carefully revised the manuscript.

Chapter 4 is published as Hongxin Wei, Renchunzi Xie, Lei Feng, Bo Han, Bo An. Deep Learning from Multiple Noisy Annotators as A Union. IEEE Transactions on Neural Networks and Learning Systems (TNNLS), DOI: 10.1109/TNNLS.2022.3168696. The contributions of the co-authors are as follows:

- I proposed the improved solutions, conducted the experiments, and wrote the main manuscript.
- Dr. Lei Feng provided the initial project direction and revised the manuscript drafts.
- Renchunzi Xie assisted in conducting the experiments.
- Prof. Bo An and Prof. Bo Han provided insightful comments and carefully revised the manuscript.

Chapter 5 is published as Hongxin Wei, Renchunzi Xie, Hao Cheng, Lei Feng, Bo An, Yixuan Li. Mitigating Neural Network Overconfidence with Logit Normalization. Proceedings of the 39th International Conference on Machine Learning (ICML). 2022. The contributions of the co-authors are as follows:

- I proposed the idea and the solution.
- I conducted the experiments and wrote the main manuscript. Prof. Yixuan Li discussed with me about how to properly present the key idea of the manuscript.
- Dr. Hao Cheng assisted in conducting the experiments.
- Renchunzi Xie, Dr. Lei Feng, Prof. Bo An and Prof. Yixuan Li provided insightful comments and carefully revised the manuscript.

Chapter 6 is published as Hongxin Wei, Lue Tao, Renchunzi Xie, Bo An. Open-set Label Noise Can Improve Robustness Against Inherent Label Noise. Advances in Neural Information Processing Systems (NeurIPS). pp. 7978-7992. 2021. The contributions of the co-authors are as follows:

- I proposed the idea and the solution.
- I conducted the experiments and wrote the main manuscript. Lue Tao and Prof. Bo An discussed with me about how to properly present the key idea of the manuscript.
- Renchunzi Xie assisted in conducting the experiments.
- Lue Tao discussed with me about the part of theoretical analysis.
- Lue Tao, Renchunzi Xie and Prof. Bo An provided insightful comments and carefully revised the manuscript.

Chapter 7 is published as Hongxin Wei, Lue Tao, Renchunzi Xie, Lei Feng, Bo An. Open-Sampling: Exploring Out-of-Distribution Data for Re-balancing Long-tailed Datasets. Proceedings of the 39th International Conference on Machine Learning (ICML), 2022. The contributions of the co-authors are as follows:

- I proposed the idea and the solution.
- I conducted the experiments and wrote the main manuscript. and Prof. Bo An discussed with me about how to properly present the key idea of the manuscript.
- Renchunzi Xie assisted in conducting the experiments.
- Lue Tao discussed with me about the part of theoretical analysis.
- Lue Tao, Renchunzi Xie, Dr. Lei Feng, and Prof. Bo An provided critical comments and carefully revised the manuscript.

06/01/2023

Date

HONGXIN WEI

Acknowledgements

First and foremost, I would like to express my deepest appreciation to my supervisor, Prof. Bo An. Without the great opportunity he provided, I could not have undertaken this Ph.D. journey, which has been a turning point in my life. During these years, he always maintains a gentle, patient, and kind style. He has not only provided a singularly suitable research environment with the greatest freedom, but also taught me the necessary knowledge in my research life, such as how to be professional and how to properly communicate with others. Words cannot express my gratitude to him. The experience in his group would definitely shape my future in many dimensions. I am so fortunate to become a Ph.D. student under his supervision and I wish to be such a nice professor like him in the future.

I also had the pleasure of working with a group of talented people: Qingyu Guo, Jiarui Gan, Haipeng Chen, Mengchen Zhao, Jiuchuan Jiang, Youzhi Zhang, Xinrun Wang, Lei Feng, Aye Phyu, Rundong Wang, Wei Qiu, Jakub Černý, Runsheng Yu, Yanchen Deng, Shuxin Li, Wanqi Xue, Feifei Lin, Zhuyan Yin, Xinyu Cai, Shuo Sun, Pengdeng Li, Shenggong Ji, Shufeng Kong, Hao Cheng, Wei Tao, Yuxin Li, Renchunzi Xie, Pengjie Gu, Yewen Li, Chaojie Wang, Zhenghai Xue, Longtao Zheng, Chuqiao Zong, Sheng Zang, Molei Qin, Ying Zheng, Dong Xing, Wentao Zhang, Ruhao Jiang, Yuzhou Cao, Haochong Xia. It has been an amazing experience working in the research group, thanks to everyone here I've worked with and from whom I've learned so much.

I am also thankful to my Thesis Advisory Committee (TAC) members: Prof. Hanwang Zhang and Prof. Bihan Wen. They provided me a lot of help on my research and made my Ph.D. journey easier in NTU.

Before the COVID period, I had an exciting internship at WeBank. I would like to thank Zhouyan Feng, Jun Yao and other people I met at Shenzhen. This project would not have been possible without the support of these amazing friends.

I would wish to thank these nice people I have met during my visiting at University of Wisconsin at Madison: Prof. Yixuan Li, Yiyou Sun, Yifei Ming, Xuefeng Du, Haoyue Bai, Andrew Geng, Agam Goyal, Rheeya Uppaal, Soumya Suvra Ghosal. Thank you for your guidance/discussion/accompany/help when I was alone in Madison.

I would also like to extend my thanks to many of my friends: Lue Tao, Zhuoyi Lin, Haobo Wang, Ziqi Zhang, Xiangyu Chen, Dr. Bo Han, Prof. Huiping Zhuang, Dr. Jingfeng Zhang, Xiaobo Xia, Zhaowei Zhu, Jiaheng Wei, and so on. Their great help and accompany make these years to be a colorful experience for me.

Finally, I must express my very profound gratitude to my families, including my parents, sister and my lovely wife, for providing me with unfailing support and continuous encouragement throughout my years of study and research. Without your unceasing encouragement, support, patience and attention, this accomplishment would not have been possible. Thank you.

Abstract

Modern machine learning techniques have demonstrated their excellent capabilities in many areas. Despite the human-surpassing performance in experimental settings, many researches have revealed the vulnerability of machine learning models caused by the violation of fundamental assumptions in real-world applications. Such issues significantly hinder the applicability and reliability of machine learning. This motivates the need to preserve model performance under naturally induced data corruptions or alterations across the machine learning pipeline, which is termed "Natural Robustness". To this end, this thesis starts by investigating two naturally occurring issues: label corruption and distribution shift. Thereafter, we proceed by exploring the value of out-of-distribution data in the robustness of machine learning.

Firstly, the observed labels of training examples are assumed to be ground truth. However, labels solicited from humans can often be subject to label corruption, leading to poor generalization performance. This gives rise to the importance of robustness against label corruption, where the goal is to train a robust classifier in the presence of noisy and erroneous labels. We first investigate how the diversity among multiple networks affects the sample selection and the overfitting to label noise. For the problem of learning with multiple noisy labels, we design an end-to-end learning framework to maximize the likelihood of the union annotation information, which is not only theoretically consistent but also experimentally effective and efficient.

Secondly, classic machine learning methods are built on the i.i.d. assumption that training and testing data are independent and identically distributed. However, neural networks deployed in the open world often struggle with out-of-distribution inputs, where they produce abnormally high confidence for both in- and out-of-distribution inputs. To alleviate this issue, we first reveal why the cross-entropy loss encourages the model to be overconfident. Then, we design a simple fix to the cross-entropy loss that enhances many existing post-hoc methods for OOD

detection. Training with the proposed loss, the network tends to give conservative predictions and results in strong separability of softmax confidence scores between in- and out-of-distribution inputs.

Lastly, traditional machine learning algorithms only exploit information from in-distribution examples that are normally expensive and challenging to collect. Thus, exploring the value of out-of-distribution examples, which are almost for free, is of great importance theoretically and practically. We investigate how open-set noisy labels affect the generalization and robustness against inherent noisy labels, how to theoretically analyze the effect of open-set noisy labels from the perspective of SGD noises, and design algorithms that utilize out-of-distribution examples to improve label-noise robustness. Besides, we provide the first attempt to utilize out-of-distribution data to rebalance the class priors of long-tailed datasets and study the effect of out-of-distribution data on the learned representations in long-tailed learning.

We evaluate the effectiveness and robustness of all the introduced methods on multiple simulated and real-world benchmarks. The reported results indicate that our methods are superior to many state-of-the-art approaches for alleviating the corresponding issues. We hope our efforts provide insights to inspire specially designed methods for these robust issues, and expedite the exploration of out-of-distribution examples for designing effective and robust systems.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Machine learning is a subset of artificial intelligence that aims to learn meaningful patterns automatically from data [8]. In the past two decades, it has been developed into a commonly utilized tool for a variety of real-world tasks, which require information extraction from large-scale datasets. Machine learning algorithms are generally designed to learn a mapping function based upon a given training dataset. Specifically, the training dataset generally contains a collection of input-output pairs, where the input is a feature vector (*instance*) and the output indicates the ground-truth *label*. As a common technique of machine learning, deep learning utilizes artificial neural networks with multiple layers between the input and output layers, which is inspired by the biological neural networks in animal brains [9]. In this thesis, we focus on deep learning.

Deep learning techniques have demonstrated their excellent capabilities in many areas like computer vision [10] and natural language processing [11]. Despite the human-surpassing performance in experimental settings, many researchers have revealed the vulnerability of deep learning models caused by the violation of fundamental assumptions in real-world applications [12, 13]. Such issues significantly hinder the applicability and reliability of deep learning. Therefore, *Robust deep learning*, which aims to preserve model performance under data corruptions or alterations across the machine learning pipeline, has raised increasing attention in recent years [13]. In terms of the cause of data corruptions, there are two typical

classes in robust deep learning, *Adversarial robustness* and *Natural robustness*. The former indicates those malicious issues caused by human intervention, including *adversarial example* [12], *data poisoning* [14], *backdoor learning* [15], *model inversion* [16] and *membership inference* [17]. Conversely, *natural robustness* denotes naturally induced data corruptions, such as *label noise* [18], *distribution shift* [19], and *class imbalance* [20].

In this thesis, we focus on two naturally induced issues: *label noise* and *distribution shift*, which lie in the subareas of weakly supervised learning [21] and open-world machine learning [22] respectively. In the label noise setting, the observed label of each training example might be different from the ground truth due to task difficulty and imperfect annotators. The issue of noisy labels has been commonly observed in many real-world scenarios, such as crowdsourcing [23] and online queries [24]. Distribution shift [19, 25], where a model is deployed on a data distribution different from what it was trained on, also poses significant robustness challenges in real-world applications. Such shifts are often unavoidable in the wild and have been shown to degrade model performance substantially in applications such as robotics [26], biomedicine, wildlife conservation and criminal justice. For instance, trained models can systematically fail when tested on patients from different hospitals or people from different demographics. Ideally, inputs with semantic shift should not be predicted with high confidence and trained models are expected to generalize to inputs with only covariant shift.

1.2 Motivations

As introduced above, both *label noise* and *distribution shift* are essential issues in robust deep learning, which are increasingly important for deploying machine learning models in the open world [21, 22]. Therefore, it is of great importance to develop effective and efficient methods to improve the model performance under such issues and provide extensive analysis to deepen the understanding of them in the context of deep learning. Although many research works have been proposed on these topics, they have some limitations yet and there also exist some unresolved issues. Specifically, previous works in learning with label noise claim that the "disagreement" strategy is crucial for alleviating the overfitting issue [27–29], but this would prevent the training process from efficient use of training examples,

leading to suboptimal performance in model generalization. Besides, when it goes to the setting of crowdsource label, where each training example is provided with multiple noisy labels, previous training algorithms are proposed to deal with each noisy label separately [30]. Such framework suffers from computational inefficiency and does not hold learning consistency.

For distribution shift, previous works have revealed that neural networks generally suffer from the overconfidence issue, where they produce abnormally high confidence for both in- and out-of-distribution inputs [25, 31]. However, yet to date, the community still has a limited understanding of the fundamental cause and mitigation of the overconfidence issue. Furthermore, existing robust algorithms only utilize in-distribution data for training and typically considered out-of-distribution (OOD) examples to be harmful to the training of deep neural networks [32–35]. But it is expensive and challenging to collect the required in-distribution data in many real-world applications, while OOD data is almost for free. Therefore, exploring the value of OOD instances in robust deep learning is of great importance theoretically and practically.

The goal of this thesis is to systematically investigate how to effectively and efficiently solve the aforementioned issues.

1.3 Major Contributions

The main contributions of this thesis are summarized as follows:

- We propose an effective approach called JoCoR to improve the label-noise robustness with the agreement maximization principle, which reduces the diversity of two networks during training. It breaks the conventional wisdom that keeping disagreement among networks is critical for robust learning with noisy labels. Our method utilizes a joint loss with Co-Regularization for both the parameter updating and example selection. Empirical results demonstrate that JoCoR is superior to many state-of-the-art approaches.

- We propose a novel end-to-end deep learning method named UnionNet, which is not only theoretically consistent but also experimentally effective and efficient. We prove the learning consistency of UnionNet by establishing an

estimation error bound, which shows that obtained empirical risk minimizer by UnionNet would converge to the true risk minimizer as the amount of training data tends to infinity. Experimental results demonstrate the effectiveness of UnionNet in model generalization and training efficiency.

- We introduce LogitNorm, a simple and effective alternative to the cross-entropy loss, which decouples the influence of logits' norm from the training procedure. Empirical results show that LogitNorm can improve both OOD detection and confidence calibration while maintaining the classification accuracy on in-distribution data.

- We provide the first attempt to improve learning with noisy labels with out-of-distribution data. In particular, we proposed Open-set regularization with Dynamic Noisy Labels (ODNL), a novel technique that enhances many existing robust training methods for label-noise robustness. We show that ODNL can be considered as a variant of SGD noises that usually leads to a flat minimum. Extensive experiments show that ODNL can help improve the robustness against various types of noisy labels and improve the OOD detection performance.

- we propose a simple yet effective method termed Open-sampling, by introducing OOD instances to re-balance the class priors of the training dataset. Open-sampling is the first to utilize OOD instances in long-tailed learning. Open-sampling not only re-balances the training dataset, but also promotes the neural network to learn more separable representations. Besides, we also present a guideline about the open-set auxiliary datasets. Experiments validate the effectiveness of Open-sampling in both model generalization and OOD detection.

1.4 Thesis Organization

The remaining part of this thesis is organized as below. Chapter 2 introduces preliminary knowledge and related literature of this thesis. Chapter 3 presents an effective approach with agreement maximization principle for learning with noisy labels. Chapter 4 presents a theoretically consistent method for learning with

multiple noisy labels. Chapter 5 introduces a simple and effective fix to the cross-entropy loss to mitigate the overconfidence issue for OOD detection. Chapter 6 explores the benefits of out-of-distribution data to improve label-noise robustness. Chapter 7 presents a novel method to introduce OOD instances to improve robustness against class imbalance. Chapter 8 concludes this thesis and highlights possible directions of future work.

Chapter 2

Preliminaries

2.1 Background

2.1.1 Supervised learning

Supervised learning is a typical machine learning paradigm, where collected data consists of labelled examples [8]. Each training example is composed of a feature vector (*instance*) and a label. The goal of supervised learning is to learn a mapping function that reflects input vectors to output labels, based upon a large amount of training examples.

In traditional supervised learning, there exist some fundamental assumptions that are essential to secure the generalization performance of machine learning models [8, 9]. One important assumption is that the observed labels of training examples are generally assumed to be perfect, i.e., ground truth. Most of the successful deep learning algorithms are heavily dependent on ground-truth labels that are correctly assigned to the training dataset. However, it is usually challenging to collect such high-quality datasets with strong supervision information in many real-world scenarios [24, 27].

Another common assumption of supervised learning is the i.i.d. assumption, where training and testing data are independent and identically distributed [8]. In other words, training data and test data are assumed to be sampled from the same distribution, which we usually call true distribution. Nevertheless, neural networks

deployed in the open world often struggle with out-of-distribution inputs, which are far away from the true distribution [25]. In this thesis, we focus on these robustness issues that violate the fundamental assumptions of supervised learning.

Here, we first formulate the problem of multi-class classification, where the size of the label space is larger than 2. Let $\mathcal{X} \subset \mathbb{R}^d$ be the feature space and $\mathcal{Y} = \{1, \ldots, K\}$ be the label space, we suppose the training dataset with N examples is given as $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$, where $\boldsymbol{x}_i \in \mathcal{X}$ is the i-th instance sampled from a certain data-generating distribution $\mathcal{P}_{\mathcal{X}\mathcal{Y}}$ in an i.i.d. manner and $y_i \in \{1, \ldots, K\}$ represents its observed label. A classifier is a function that maps from the input space to the label space $f : \mathcal{X} \to \mathbb{R}^K$ with trainable parameters $\boldsymbol{\theta} \in \mathbb{R}^p$. Then, the goal of multi-class classification is to train a multi-class classifier $f : \mathcal{X} \to \mathbb{R}^K$ that minimizes the following classification risk:

$$\mathcal{R}(f) = \mathbb{E}_{\mathcal{P}_{\mathcal{X}\mathcal{Y}}} \left[\mathcal{L}(f(\boldsymbol{x}), y)) \right], \tag{2.1}$$

where $\mathbb{E}_{\mathcal{P}_{\mathcal{X}\mathcal{Y}}} [\,\cdot\,]$ denotes the expectation over the joint distribution $\mathcal{P}_{\mathcal{X}\mathcal{Y}}$ and $\mathcal{L} : \mathbb{R}^K \times \mathcal{Y} \to \mathbb{R}_+$ is a multi-class loss function that measures how well a classifier fits a labelled example.

Generally, the underlying joint distribution $\mathcal{P}_{\mathcal{X}\mathcal{Y}}$ is unknown, so we need to approximate the classification risk in (2.1) based upon those given training examples $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$. With the conventional i.i.d. assumption, the following empirical risk estimator is unbiased:

$$\tilde{\mathcal{R}}(f) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(f(\boldsymbol{x}_i), y_i). \tag{2.2}$$

One of the commonly used loss function for multi-class classification is the cross-entropy loss, which is also known as the log loss. Formally, the cross-entropy loss with the softmax activation function is calculated using the following formula:

$$\mathcal{L}_{\mathrm{CE}}(f(\boldsymbol{x}_i), y_i) = -\log p(y|\boldsymbol{x}) = -\log \frac{e^{f_y(\boldsymbol{x})}}{\sum_{i=1}^{k} e^{f_i(\boldsymbol{x})}}. \tag{2.3}$$

where $f_y(\boldsymbol{x})$ denotes the y-th element of $f(\boldsymbol{x})$ corresponding to the ground-truth label y, and $p(y|\boldsymbol{x})$ is the corresponding softmax probability. As described before,

these fundamental assumptions, such as i.i.d. assumption and true-label assumption, are generally challenging to maintain in real-world scenarios. In the following, we will introduce some related subareas of machine learning, which aims to reduce the dependence of learning algorithms on these assumptions.

2.1.2 Weakly supervised learning

In general, traditional supervised learning assumes that training examples $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$, composed of pairs of an instance and its ground-truth label, are identically and independently sampled from the true distribution $\mathcal{P}_{\mathcal{X}\mathcal{Y}}$. However, it is usually challenging to obtain the true label of the given instance. Therefore, weakly supervised learning aims to train a classifier based upon a training dataset where the instances are not necessarily paired with their true labels [21]. In this area, various weakly supervised learning frameworks have been extensively investigated, such as *semi-supervised learning* [36], *noisy-label learning* [37, 38], *positive-unlabeled learning* [39], *partial-label learning* [40, 41], and *complementary-label learning* [42].

For example, semi-supervised learning concerns the situation where a small amount of labeled data and abundant unlabeled data are available [36]. Here, only exploring those labeled data is insufficient to train a good model. Formally, this task is to learn a classifier $f : \mathcal{X} \to \mathbb{R}^K$ from a training dataset $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m), \boldsymbol{x}_{m+1}, \boldsymbol{x}_{m+2}, \ldots, \boldsymbol{x}_n\}$, which includes m labeled training examples (i.e., those paired with y_i) and $n - m$ unlabeled instances. Generally, labeled data and unlabeled data are assumed to be sampled from the same distribution.

Another typical class of weakly supervised learning is complementary-label learning [42], where each training example is paired with a single complementary label. In particular, the complementary label only specifies one of the classes that the training example does not belong to. Formally, we denote the complementarity labeled examples as $\{\boldsymbol{x}_i, \tilde{y}_i\}_{i=1}^N$, where $\tilde{y}_i \in \mathcal{Y}$ is a complementary label of the instance $\boldsymbol{x}_i$. In particular, each example with complementary label is independently sampled from $\tilde{\mathcal{P}}_{\mathcal{X}\mathcal{Y}}$.

2.1.3 Open-world machine learning

In traditional machine learning, a common assumption is that training data and testing data are sampled from the same distribution. Consequently, a trained model can only predict with the input instances that are similar to the instances in the training. However, in a real-world scenario, the trained model is generally deployed in interactive and automated applications, which are required to work in dynamic environment, e.g., chatbot [43] and self-driving car [44]. In such cases, the model may need to deal with some inputs that are never seen before, but it cannot address that kind of situation due to the close-world assumption. To this end, open-world machine learning is designed to enable the machine to learn as a human being in a dynamic environment [22].

Open-world machine learning is different from the aforementioned semi-supervised learning [36], where a few labeled data and large number of unlabeled data are provided in the training stage. Semi-supervised learning still follows the closed world assumption, since the unlabeled data generally belongs to existing classes and is sampled from the same distribution as the labeled data. Recently, *open-world semi-supervised learning* [45, 46] is proposed to formalize the setting that unlabeled data and testing data may contain instances that belong to unseen classes. For each unlabeled data, the model is designed to assign either one of the previously seen classes or a novel class. In this way, the new paradigm is able to handle the open-world setting, where a trained model needs to deal with examples from unseen classes.

Another related learning task is *transfer learning* [47, 48], which aims to use knowledge transfer and fine-tuning to make prediction for testing data from new domains. While open-world machine learning is primarily concerned with models that can handle new and previously unseen data, transfer learning focuses on improving the performance of models on a specific task by reusing knowledge from related tasks.

In this part, we have revisited the notion of supervised learning, and provide a discussion about two related subareas of machine learning, including weakly supervised learning and open-world machine learning. With this background, we proceed by presenting several advanced topics in robust deep learning.

2.2 Natural Robustness of Machine Learning

Deep learning techniques have demonstrated their excellent capabilities in many areas like computer vision [10] and natural language processing [11]. Despite the human-surpassing performance in experimental settings, many researches have revealed the vulnerability of machine learning model caused by the violation of fundamental assumptions in real-world applications [12, 13]. Such issues significantly hinder the applicability and reliability of machine learning. In this thesis, we focus on *Natural Robustness*, which concerns the model robustness under naturally induced data corruptions or alterations across the machine learning pipeline. In this section, we will introduce two training robustness issues, including label noise and class imbalance, as well as testing robustness against distribution shift.

2.2.1 Training robustness against label noise

Most deep neural networks are trained in a supervised manner, which heavily relies on numerous training instances with accurate labels. However, collecting large-scale datasets with fully precise annotations is expensive and time-consuming. To alleviate this problem, data annotation companies choose some alternating methods such as crowdsourcing [23, 29] and online queries [24] to improve labelling efficiency. Unfortunately, these methods usually suffer from unavoidable noisy labels, which have been shown to lead to noticeable decrease in performance of DNNs [49]. As deep neural networks have the high capacity to fit noisy labels [50], it is challenging to train deep networks robustly with noisy labels.

For the setting of learning with noisy labels, we consider multi-class classification with K classes. Let $\mathcal{X} \subset \mathbb{R}^d$ be the feature space and $\mathcal{Y} = \{1, \dots, K\}$ be the label space, we suppose the training dataset with N examples is given as $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$, where $\boldsymbol{x}_i \in \mathcal{X}$ is the i-th instance sampled from the true distribution in an i.i.d. manner and $y_i \in \{1, \dots, K\}$ represents its observed label that might be different from the ground truth. A classifier is a function that maps the feature space to the label space $f : \mathcal{X} \to \mathbb{R}^K$ with trainable parameter $\boldsymbol{\theta} \in \mathbb{R}^p$. Generally, we consider the common case where the function f is a DNN with the softmax output layer. The objective function of standard training can be represented as a cross-entropy

loss:

$$\mathcal{L}_{CCE} = \mathbb{E}_{\mathcal{P}_{\mathcal{X}\mathcal{Y}}} \left[\ell \left(f(\boldsymbol{x}; \boldsymbol{\theta}), y \right) \right] \simeq -\frac{1}{N} \sum_{i=1}^{N} \boldsymbol{e}^{y_i} \log f \left(\boldsymbol{x}_i; \boldsymbol{\theta} \right), \tag{2.4}$$

where $f(\boldsymbol{x}_i; \boldsymbol{\theta})$ denotes the output of the classifier f with $\boldsymbol{\theta}$ for $\boldsymbol{x}_i$ and $\boldsymbol{e}^{y_i} := (0, \ldots, 1, \ldots, 0)^\top \in \{0, 1\}^K$ is a one-hot vector and only the y_i-th entry of $\boldsymbol{e}^{y_i}$ is 1.

Another common setting is learning with crowdsource labels, where each training example is provided with multiple noisy labels [51]. Let $\mathcal{D} = \{(\boldsymbol{x}_i, Y_i)\}_{i=1}^n$ be a crowdsourced dataset that includes n examples with k classes, where for each instance $\boldsymbol{x}_i \in \mathbb{R}^d$, we receive a set of crowdsourced labels $Y_i = \{\tilde{y}_i^j \mid j = 1 \ldots m\}$, with $\tilde{y}_i^j$ representing the label provided by the j-th annotator in a set of m annotators. It is worth noting that the number of crowdsourced labels of different examples could be different. Hence, $|Y_i|$ may not be equal to $|Y_j|$ if $i \neq j$. Following [23, 30, 52, 53], we also assume each instance $\boldsymbol{x}_i$ has its corresponding latent ground-truth label y_i and it is related to the crowdsourced labels Y_i. It is worth noting that learning with crowdsourced data is a multi-class classification problem, which means there is only a true label for each training instance. This setting is different from multi-label learning [54–58] where multiple true labels are provided for each each training instance. Under this setting, our goal is to train a classifier f based on a crowdsourced dataset $\mathcal{D} = \{(\boldsymbol{x}_i, Y_i)\}_{i=1}^n$ that could accurately predict the ground-truth label of an unseen instance in the test phase.

Learning with noisy labels is a vital topic in weakly-supervised learning, attracting much recent interest with several directions. 1) Some methods propose to train on selected examples, using small-loss selection [27, 59, 60], GMM distribution [61, 62] or (dis)agreement between two models [28, 59, 60]. 2) Some methods aim to design sample weighting schemes that give higher weights on clean examples [37, 63–65]. 3) Loss correction is also a popular direction based on an estimated noise transition matrix [66, 67] or the model's predictions [61, 68–71]. 4) Robust loss functions are also studied to have a theoretical guarantee [72–76]. 5) Some methods apply regularization techniques to improve generalization under the settings of label noise [77], like gradient clipping [78], label smoothing [79, 80], temporal ensembling [81] and virtual adversarial training [82]. 6) Some training strategies for combating noisy labels are built based upon semi-supervised learning methods [62, 83]. Studying robustness against label noise helps us further understand the essence of training deep neural networks.

2.2.2 Training robustness against class imbalance

The success of deep neural networks (DNNs) heavily relies on large-scale datasets with balanced distribution [84, 85]. However, in real-world applications like autonomous driving and medical diagnosis, large-scale datasets naturally exhibit imbalanced and long-tailed distributions, i.e., a few classes (majority classes) occupy most of the data while most classes (minority classes) are under-represented [86–88]. It has been shown that training on long-tailed datasets leads to poor generalization performance, especially on the minority classes [89–91]. Thus, designing effective algorithms to improve robustness against class imbalance is of great practical importance.

Long-tailed learning can be regarded as a challenging sub-task of class-imbalanced learning [92]. The dominant distinction is that the classes of long-tailed learning follow a long-tailed class distribution, which is not necessary for class-imbalanced learning. More differences include that in long-tailed learning the number of classes is usually large, and the tail-class examples are often very scarce, whereas the number of minority-class examples in class-imbalanced learning is not necessarily small in an absolute sense. Despite these differences, both seek to resolve the class imbalance, so some high-level solution ideas (e.g., class re-balancing) are shared between them.

In this task, we consider a multi-class classification problem, where the input space is denoted by $\mathcal{X} \in \mathbb{R}^d$ and the label space $\mathcal{Y}$ is $\{1, \ldots, K\}$. We denote by $\mathcal{D}_{\text{train}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N \in \mathcal{X} \times \mathcal{Y}$ the training dataset with N examples. Let n_j be the number of instances in class j, then $N = \sum_{j=1}^K n_j$. Let $P_{\text{s}}(X, Y)$ define the underlying training (source) distribution and $P_{\text{t}}(X, Y)$ define the test (target) distribution. Generally, the class imbalance problem assumes that the test data have the same class conditional probability as the training data, i.e., $P_{\text{s}}(X|Y) = P_{\text{t}}(X|Y)$, while their class priors are different, i.e., $P_{\text{s}}(Y) \neq P_{\text{t}}(Y)$.

A conventional idea for learning with long-tailed data is Re-sampling, which aims to re-balance the class priors of the training dataset. Under-sampling methods remove examples from the majority classes, which is infeasible under extremely data imbalanced settings [93, 94]. The over-sampling method adds repeated examples

for the minority classes, usually causing over-fitting to the minority classes [95–97]. Some methods utilize synthesized in-distribution examples to alleviate the over-fitting issue but introduce extra noise [6, 98, 99].

Another classic algorithm is Re-weighting, which proposes to assign adaptive weights for different classes or examples. Generally, the vanilla scheme re-weights classes proportionally to the inverse of their frequency [100]. Focal loss [101] assigns low weights to the well-classified examples. Class-balanced loss [102] proposes to re-weight by the inverse effective number of examples. However, these re-weighting methods tend to make the optimization of DNNs difficult under extremely data imbalanced settings [103, 104].

In addition to the data re-balancing approaches, some other solutions are also applied for class-imbalanced learning, including transfer-learning based methods [105, 106], two-stage training methods[91, 107–109], training objective based methods [7, 110, 111], expert methods [20] and self-supervised learning methods [112, 113]. For example, transfer-learning based methods addressed the class-imbalanced issue by transferring features learned from head classes with abundant training instances to under-represented tail classes [105, 106]. Two-stage training methods proposed to apply decoupled training where the classifier is re-balanced during the fine-tuning stage [91, 107–109].

2.2.3 Testing robustness against distribution shift

Most existing machine learning models are trained based on the closed-world assumption, where the test data is assumed to be drawn i.i.d. from the same distribution as the training data, known as in-distribution (ID). However, when models are deployed in an open-world scenario, test examples can be out-of-distribution (OOD) and therefore should be handled with caution. The distributional shifts can be caused by semantic shift (e.g., OOD examples are drawn from different classes) [25, 109], or covariate shift (e.g., OOD examples from a different domain) [19]. In addition, target and conditional shifts are also investigated in domain adaptation [114]. A reliable learning system should not only produce accurate predictions on known context, but also detect unknown examples and reject them. For instance, a well-trained food classifier should be able to detect non-food images such as selfies uploaded by users, and reject such input instead of blindly classifying them into

existing food categories. In safety-critical applications such as autonomous driving, the driving system must issue a warning and hand over the control to drivers when it detects unusual scenes or objects it has never seen during training. This gives rise to the importance of out-of distribution (OOD) detection, which determines whether an input is in-distribution (ID) or OOD.

We consider a supervised multi-class classification problem. We denote by $\mathcal{X}$ the input space and $\mathcal{Y} = \{1, \dots, K\}$ the label space with K classes. The training dataset $\mathcal{D}_{\text{train}} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{N}$ consists of N data points, sampled *i.i.d.* from a joint data distribution $\mathcal{P}_{\mathcal{X}\mathcal{Y}}$. We use $\mathcal{P}_{\text{in}}$ to denote the marginal distribution over $\mathcal{X}$, which represents the in-distribution (ID). Given the training dataset, we learn a classifier $f : \mathcal{X} \to \mathbb{R}^K$ with trainable parameter $\theta \in \mathbb{R}^p$, which maps an input to the output space. An ideal classifier can be obtained by minimizing the following expected risk:

$$\mathcal{R}_{\mathcal{L}}(f) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{P}_{\mathcal{X}\mathcal{Y}}} \left[\mathcal{L}\left(f(\boldsymbol{x};\theta), y\right) \right],$$

where $\mathcal{L}$ is the commonly used cross-entropy loss with the softmax activation function:

$$\mathcal{L}_{\text{CE}}(f(\boldsymbol{x};\theta), y) = -\log p(y|\boldsymbol{x}) = -\log \frac{e^{f_y(\boldsymbol{x};\theta)}}{\sum_{i=1}^{k} e^{f_i(\boldsymbol{x};\theta)}}.$$

Here, $f_y(\boldsymbol{x};\theta)$ denotes the y-th element of $f(\boldsymbol{x};\theta)$ corresponding to the ground-truth label y, and $p(y|\boldsymbol{x})$ is the corresponding softmax probability.

During the deployment time, it is ideal that the test data are drawn from the same distribution $\mathcal{P}_{\text{in}}$ as the training data. However, in reality, inputs from unknown distributions can arise, whose label set may have no intersection with $\mathcal{Y}$. Such inputs are termed out-of-distribution (OOD) data and should not be predicted by the model.

The OOD detection task can be formulated as a binary classification problem: determining whether an input $\boldsymbol{x} \in \mathcal{X}$ is from $\mathcal{P}_{\text{in}}$ (ID) or not (OOD). OOD detection can be performed by a level-set estimation:

$$g(\boldsymbol{x}) = \begin{cases} \text{in} & \text{if } S(\boldsymbol{x}) \geq \gamma \\ \text{out} & \text{if } S(\boldsymbol{x}) < \gamma \end{cases}, \tag{2.5}$$

where $S(\boldsymbol{x})$ denotes a scoring function and γ is commonly chosen so that a high fraction (e.g., 95%) of ID data is correctly classified. By convention, examples with higher scores $S(\boldsymbol{x})$ are classified as ID and vice versa. In addition to the MSP score, there are a variety of popular OOD scoring functions including MSP [115], ODIN [116], energy score [31] and GradNorm [117].

OOD detection is an increasingly important topic for deploying machine learning models in the open world and has attracted a surge of interest in two directions. 1) Some methods aim to design scoring functions for OOD detection, such as OpenMax score [118], maximum softmax probability [115], ODIN score [116, 119], Mahalanobis distance-based score [120], Energy-based score [31, 121, 122], ReAct [123], GradNorm score [117], and non-parametric KNN-based score [124, 125]. 2) Some works address the out-of-distribution detection problem by training-time regularization [3, 31, 126–135]. For example, models are encouraged to give predictions with uniform distribution [3, 126] or higher energies [31, 135–138] for outliers. The energy-based regularization has a direct theoretical interpretation as shaping the log-likelihood, hence naturally suits OOD detection. Contrastive learning methods are also employed for the OOD detection task [139–141], which can be computationally expensive.

Chapter 3

Combating Noisy Labels with Agreement Maximization

3.1 Motivation

Deep Neural Networks (DNNs) achieve remarkable success on various tasks, and most of them are trained in a supervised manner, which heavily relies on a large number of training instances with accurate labels [4]. However, collecting large-scale datasets with fully precise annotations is expensive and time-consuming. To alleviate this problem, data annotation companies choose some alternating methods such as crowdsourcing [23, 29] and online queries [24] to improve labeling efficiency. Unfortunately, these methods usually suffer from unavoidable noisy labels, which have been proven to lead to noticeable decrease in the performance of DNNs [49, 50].

As this problem has severely limited the expansion of neural network applications, a large number of algorithms have been developed for learning with noisy labels, which belongs to the family of weakly supervised learning frameworks [142–148]. Some of them focus on improving the methods to estimate the latent noisy transition matrix [64, 149, 150]. However, it is challenging to estimate the noise transition matrix accurately. An alternative approach is training on selected or weighted samples, e.g., Mentornet [63], gradient-based reweight [65] and Co-teaching [27]. Furthermore, the state-of-the-art methods including Co-teaching+ [60] and Decoupling [28] have shown excellent performance in learning with noisy labels by introducing the "Disagreement" strategy, where "when to update" depends on a

disagreement between two different networks. However, there are only a part of training examples that can be selected by the "Disagreement" strategy, and these examples cannot be guaranteed to have ground-truth labels [27]. Therefore, there arises a question to be answered: Is "Disagreement" necessary for training two networks to deal with noisy labels?

Motivated by Co-training for multi-view learning and semi-supervised learning that aims to maximize the agreement on multiple distinct views [151–154], a straightforward method for handling noisy labels is to apply the regularization from peer networks when training each single network. However, although the regularization may improve the generalization ability of networks by encouraging agreement between them, it still suffers from memorization effects on noisy labels [50]. To address this problem, we propose a novel approach named JoCoR (**Jo**int Training with **Co-R**egularization). Specifically, we train two networks with a joint loss, including the conventional supervised loss and the Co-Regularization loss. Furthermore, we use the joint loss to select small-loss examples, thereby ensuring the error flow from the biased selection would not be accumulated in a single network.

To show that JoCoR [1] significantly improves the robustness of deep learning on noisy labels, we conduct extensive experiments on both simulated and real-world noisy datasets, including MNIST, CIFAR-10, CIFAR-100 and Clothing1M datasets. Empirical results demonstrate that the robustness of deep models trained by our proposed approach is superior to many state-of-the-art approaches. Furthermore, the ablation studies clearly demonstrate the effectiveness of Co-Regularization and Joint Training.

3.2 The Proposed Approach

As mentioned before, we suggest to apply the agreement maximization principle to tackle the problem of noisy labels. In our approach, we encourage two different classifiers to make predictions closer to each other by explicit regularization method instead of hard sampling employed by the "Disagreement" strategy. This method could be considered as a meta-algorithm that trains two base classifiers by one loss

[1] The work in this chapter has been published in [59].

function, which includes a regularization term to reduce divergence between the two classifiers.

For multi-class classification with M classes, we suppose the dataset with N samples is given as $D = \{\boldsymbol{x}_i, y_i\}_{i=1}^{N}$, where $\boldsymbol{x}_i$ is the i-th instance with its observed label as $y_i \in \{1, \ldots, M\}$. Similar to Decoupling and Co-teaching+, we formulate the proposed JoCoR approach with two deep neural networks denoted by $f(\boldsymbol{x}, \boldsymbol{\Theta}_1)$ and $f(\boldsymbol{x}, \boldsymbol{\Theta}_2)$, while $\boldsymbol{p}_1 = [p_1^1, p_1^2, \ldots, p_1^M]$ and $\boldsymbol{p}_2 = [p_2^1, p_2^2, \ldots, p_2^M]$ denote their prediction probabilities of instance $\boldsymbol{x}_i$, respectively. In other words, $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ are the outputs of the "softmax" layer in $\boldsymbol{\Theta}_1$ and $\boldsymbol{\Theta}_2$.

Network. For JoCoR, each network can be used to predict labels alone, but during the training stage these two networks are trained with a pseudo-siamese paradigm, which means their parameters are different but updated simultaneously by a joint loss (see Figure 3.1). In this work, we call this paradigm as "Joint Training".

Specifically, our proposed loss function ℓ on $\boldsymbol{x}_i$ is constructed as follows:

$$\ell(x_i) = (1 - \lambda) * \ell_{\text{sup}}(\boldsymbol{x}_i, y_i) + \lambda * \ell_{\text{con}}(\boldsymbol{x}_i). \tag{3.1}$$

In the loss function, the first part ℓ_{sup} is conventional supervised learning loss of the two networks, the second part ℓ_{con} is the contrastive loss between predictions of the two networks for achieving Co-Regularization.

Classification loss. For multi-class classification, we use Cross-Entropy Loss as the supervised part to minimize the distance between predictions and labels.

$$\begin{aligned} \ell_{\text{sup}}(\boldsymbol{x}_i, y_i) &= \ell_{\text{C1}}(\boldsymbol{x}_i, y_i) + \ell_{\text{C2}}(\boldsymbol{x}_i, y_i) \\ &= -\sum_{i=1}^{N} \sum_{m=1}^{M} y_i \log(p_1^m(\boldsymbol{x}_i)) \\ &\quad -\sum_{i=1}^{N} \sum_{m=1}^{M} y_i \log(p_2^m(\boldsymbol{x}_i)). \end{aligned} \tag{3.2}$$

Intuitively, two networks can filter different types of errors brought by noisy labels since they have different learning abilities. In Co-teaching [27], when the two networks exchange the selected small-loss instances in each mini-batch data, the error flows can be reduced by peer networks mutually. By virtue of the joint-training paradigm, our JoCoR would consider the classification losses from both two networks during the "small-loss" selection stage. In this way, JoCoR can share

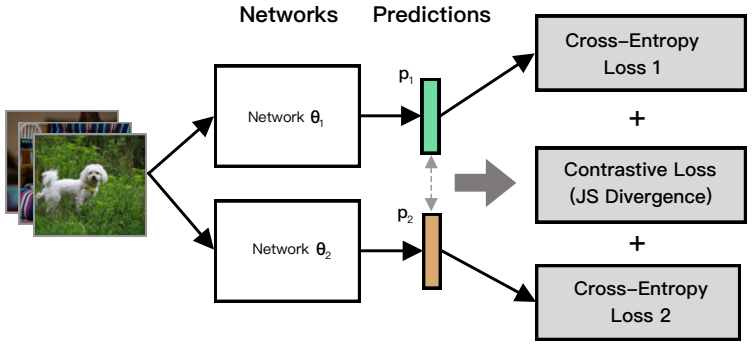


FIGURE 3.1: JoCoR schematic.

the same advantage of the cross-update strategy in Co-teaching. This argument will be clearly supported by the ablation study in the later section.

Contrastive loss. From the view of *agreement maximization principles* [151, 153], different models would agree on labels of most examples, and they are unlikely to agree on incorrect labels. Based on this observation, we apply the Co-Regularization method to maximize the agreement between two classifiers. On one hand, the Co-Regularization term could help our algorithm select examples with clean labels since an example with small Co-Regularization loss means that two networks reach an agreement on its predictions. On the other hand, the regularization from peer networks helps the model find a much wider minimum, which is expected to provide better generalization performance [154].

In JoCoR, we utilize the contrastive term as Co-Regularization to make the networks guide each other. To measure the match of the two networks' predictions $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$, we adopt the Jensen-Shannon (JS) Divergence. To simplify implementation, we could use the symmetric Kullback-Leibler(KL) Divergence to surrogate this term.

$$\ell_{\text{con}} = D_{\text{KL}}(\boldsymbol{p}_1||\boldsymbol{p}_2) + D_{\text{KL}}(\boldsymbol{p}_2||\boldsymbol{p}_1), \tag{3.3}$$

where

$$D_{\text{KL}}(\boldsymbol{p}_1||\boldsymbol{p}_2) = \sum_{i=1}^{N}\sum_{m=1}^{M} p_1^m(\boldsymbol{x}_i)\log\frac{p_1^m(\boldsymbol{x}_i)}{p_2^m(\boldsymbol{x}_i)},$$
$$D_{\text{KL}}(\boldsymbol{p}_2||\boldsymbol{p}_1) = \sum_{i=1}^{N}\sum_{m=1}^{M} p_2^m(\boldsymbol{x}_i)\log\frac{p_2^m(\boldsymbol{x}_i)}{p_1^m(\boldsymbol{x}_i)}.$$

Small-loss Selection Before introducing the details, we first clarify the connection between small losses and clean instances. Intuitively, small-loss examples are likely to be the ones that are correctly labelled [27, 65]. Thus, if we train our classifier

Algorithm 1: JoCoR

Input: Network f with $\mathbf{\Theta} = \{\mathbf{\Theta}_1, \mathbf{\Theta}_2\}$, learning rate η, fixed τ, epoch T_k and $T_{\max}$, iteration $I_{\max}$;

1: **for** $t = 1,2,\ldots,T_{\max}$ **do**
2: **Shuffle** training set D;
3: **for** $n = 1, \ldots, I_{\max}$ **do**
4: **Fetch** mini-batch D_n from D;
5: $\boldsymbol{p}_1 = f(\boldsymbol{x}, \mathbf{\Theta}_1), \forall \boldsymbol{x} \in D_n$;
6: $\boldsymbol{p}_2 = f(\boldsymbol{x}, \mathbf{\Theta}_2), \forall \boldsymbol{x} \in D_n$;
7: **Calculate** the joint loss ℓ by (3.1) using $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$;
8: **Obtain** small-loss sets $\tilde{D}_n$ by (3.4) from D_n;
9: **Obtain** L by (3.5) on $\tilde{D}_n$;
10: **Update** $\mathbf{\Theta} = \mathbf{\Theta} - \eta \nabla L$;
11: **end for**
12: **Update** $R(t) = 1 - \min\left\{\frac{t}{T_k}\tau, \tau\right\}$
13: **end for**

Output: $\mathbf{\Theta}_1$ and $\mathbf{\Theta}_2$

only using small-loss instances in each mini-batch data, it would be resistant to noisy labels.

To handle noisy labels, we apply the "small-loss" criterion to select "clean" instances (step 8 in Algorithm 1). Following the setting of Co-teaching+, we update $R(t)$ (step 12), which controls how many small-loss data should be selected in each training epoch. At the beginning of training, we keep more small-loss data (with a large $R(t)$) in each mini-natch since deep networks would fit clean data first [49, 50]. With the increase of epochs, we reduce $R(t)$ gradually until reaching $1-\tau$, keeping fewer examples in each mini-batch. Such operation will prevent deep networks from over-fitting noisy data [27].

In our algorithm, we use the joint loss (3.1) to select small-loss examples. Intuitively, an instance with small joint loss means that both two networks could be easy to reach a consensus and make correct predictions on it. As two networks have different learning abilities based on different initial conditions, the selected small-loss instances are more likely to be with clean labels than those chosen by a single model. Specifically, we conduct small-loss selection as follows:

$$\tilde{D}_n = \arg\min_{D'_n : |D'_n| \geq R(t)|D_n|} \ell\left(D'_n\right). \tag{3.4}$$

TABLE 3.1: Comparison of state-of-the-art and related techniques with our JoCoR approach. In the first column, "small loss": regarding small-loss samples as "clean" samples, which is based on the memorization effects of deep neural networks; "double classifiers": training two classifiers simultaneously; "cross update": updating parameters in a cross manner instead of a parallel manner; "joint training": training two classifiers with a joint loss; "disagreement": updating two classifiers on disagreed examples during the entire training epochs; "agreement": maximizing the agreement of two classifiers by regularization during the whole training epochs.

	Decoupling	Co-teaching	Co-teaching+	JoCoR
small loss	✗	✓	✓	✓
cross update	✗	✓	✓	✗
joint training	✗	✗	✗	✓
disagreement	✓	✗	✓	✗
agreement	✗	✗	✗	✓

After obtaining the small-loss instances, we calculate the average loss on these examples for further backpropagation:

$$L = \frac{1}{|\tilde{D}|} \sum\nolimits_{\boldsymbol{x} \in \tilde{D}} \ell(\boldsymbol{x}). \tag{3.5}$$

Relations to other approaches. We compare JoCoR with other related approaches in Table 3.1. Specifically, Decoupling applies the "disagreement" strategy to select instances while Co-teaching use small-loss criterion. Besides, Co-teaching updates parameters of networks by the "cross-update" strategy to reduce the accumulated error flow. Combining the "disagreement" strategy and the "cross-update" strategy, Co-teaching+ achieves excellent performance. As for our JoCoR, we also select small-loss examples but update the networks by Joint Training. Furthermore, we use the Co-Regularization to maximize agreement between the two networks. Note that Co-Regularization in our proposed method and "disagreement" strategy in Decoupling are both essentially to reduce the divergence between the two classifiers. The difference between them lies in that the former uses an explicit regularization methods with all training examples while the latter employs hard sampling that reduces the effective number of training examples. It is especially important in the case of small-loss selection, because the selection would further decrease the effective number of training examples.

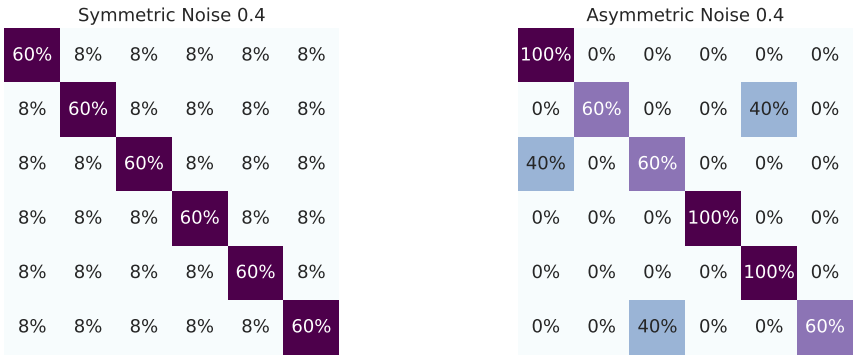


FIGURE 3.2: Example of noise transition matrix T (taking 6 classes and noise ratio 0.4 as an example)

3.3 Experiments

In this section we first compare JoCoR with some state-of-the-art approaches, then analyze the impact of Joint Training and Co-Regularization by ablation study. We also analyze the effect of λ in (3.1) by sensitivity analysis and put it in supplementary materials. Code is available at https://github.com/hongxin001/JoCoR.

3.3.1 Experiment setup

Datasets. We verify the effectiveness of our proposed algorithm on four benchmark datasets: *MNIST*, *CIFAR-10*, *CIFAR-100* and *Clothing1M* [155]. These datasets are popularly used for the evaluation of learning with noisy labels in previous literatures [69, 156, 157]. Especially, *Clothing1M* is a large-scale real-world dataset with noisy labels, which is widely used in the related works[67, 155, 158, 159]. The detailed characteristics of these datasets are shown in Table 3.2.

TABLE 3.2: Summary of datasets used in the experiments.

	# of training	# of test	# of class	size
MNIST	60,000	10,000	10	28×28
CIFAR-10	50,000	10,000	10	32×32
CIFAR-100	50,000	10,000	100	32×32
Clothing1M	1,000,000	10,000	14	224×224

Since all datasets are clean except *Clothing1M*, following [67, 69], we need to corrupt these datasets manually by the label transition matrix Q, where $Q_{ij} = \Pr[\tilde{y} = j|y = i]$ given that noisy $\tilde{y}$ is flipped from clean y. Assume that the matrix Q has

two representative structures: (1) Symmetry flipping [160]; (2) Asymmetry flipping [67]: simulation of fine-grained classification with noisy labels, where labellers may make mistakes only within very similar classes.

Following F-correction [67], only half of the classes in the dataset are with noisy labels in the setting of asymmetric noise, so the actual noise rate in the whole dataset τ is half of the noisy rate in the noisy classes. Specifically, when the asymmetric noise rate is 0.4, it means $\tau = 0.2$. Figure 3.2 shows an example of noise transition matrix.

For experiments on *Clothing1M*, we use the 1M images with noisy labels for training, the 14k and 10k clean data for validation and test, respectively. Note that we do not use the 50k clean training data in all the experiments because only noisy labels are required during the training process [70, 158]. For preprocessing, we resize the image to 256×256, crop the middle 224×224 as input, and perform normalization.

Baselines. We compare JoCoR (Algorithm 1) with the following state-of-the-art algorithms, and implement all methods with default parameters by PyTorch, and conduct all the experiments on NVIDIA Tesla V100 GPU.

- Co-teaching+ [60], which trains two deep neural networks and consists of disagreement-update step and cross-update step.
- Co-teaching [27], which trains two networks simultaneously and cross-updates parameters of peer networks.
- Decoupling [28], which updates the parameters only using instances which have different predictions from two classifiers.
- F-correction [67], which corrects the prediction by the label transition matrix. As suggested by the authors, we first train a standard network to estimate the transition matrix Q.
- As a simple baseline, we compare JoCoR with the standard deep network that directly trains on noisy datasets (abbreviated as Standard).

Network Structure and Optimizer. We use a 2-layer MLP for *MNIST*, a 7-layer CNN network architecture for *CIFAR-10* and *CIFAR-100*. The detailed

information can be found in supplementary materials. For *Clothing1M*, we use ResNet with 18 layers. The network architectures of the MLP and CNN models are shown in Table 3.3.

For experiments on *MNIST*, *CIFAR-10* and *CIFAR-100*, Adam optimizer (momentum=0.9) is used with an initial learning rate of 0.001, and the batch size is set to 128. We run 200 epochs in total and linearly decay learning rate to zero from 80 to 200 epochs.

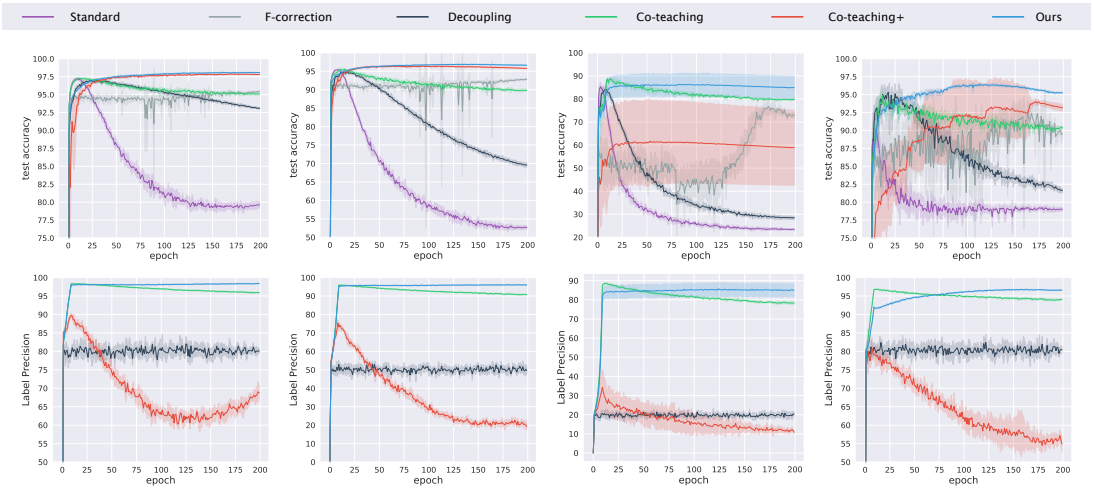
For experiments on *Clothing1M*, we also use Adam optimizer (momentum=0.9) and set batch size to 64. During the training stage, we run 15 epochs in total and set learning rate to 8×10^{-4}, 5×10^{-4} and 5×10^{-5} for 5 epochs each.

TABLE 3.3: The models used on *MNIST*, *CIFAR-10* and *CIFAR-100*

MLP on *MNIST*	CNN on *CIFAR-10*& *CIFAR-100*
28×28 Gray Image	32×32 RGB Image
Dense $28 \times 28 \longrightarrow 256$, ReLU	3×3, 64 BN, ReLU 3×3, 64 BN, ReLU 2×2 Max-pool
	3×3, 128 BN, ReLU 3×3, 128 BN, ReLU 2×2 Max-pool
	3×3, 196 BN, ReLU 3×3, 196 BN, ReLU 2×2 Max-pool
Dense $256 \longrightarrow 10$	Dense $256 \longrightarrow 100$

As for λ in our loss function (3.1), we search it in [0.05, 0.10, 0.15,. . .,0.95] with a clean validation set for best performance. When validation set is also with noisy labels, we use the small-loss selection to choose a clean subset for validation. As deep networks are highly nonconvex, even with the same network and optimization method, different initializations can lead to different local optimum. Thus, following Decoupling [28], we also take two networks with the same architecture but different initializations as two classifiers.

Measurement. To measure the performance, we use the test accuracy, i.e., *test accuracy = (# of correct predictions) / (# of test)*. Besides, we also use the label precision in each mini-batch, i.e., *label precision = (# of clean labels) / (# of all selected labels)*. Specifically, we sample $R(t)$ of small-loss instances in each mini-batch and then calculate the ratio of clean labels in the small-loss instances. Intuitively, higher label precision means less noisy instances in the mini-batch after



(A) S-20% (B) S-50% (C) S-80% (D) A-40%

FIGURE 3.3: Results on MNIST dataset. Top: test accuracy(%) vs. epochs; bottom: label precision(%) vs. epochs.

TABLE 3.4: Average test accuracy (%) on *MNIST* over the last 10 epochs.

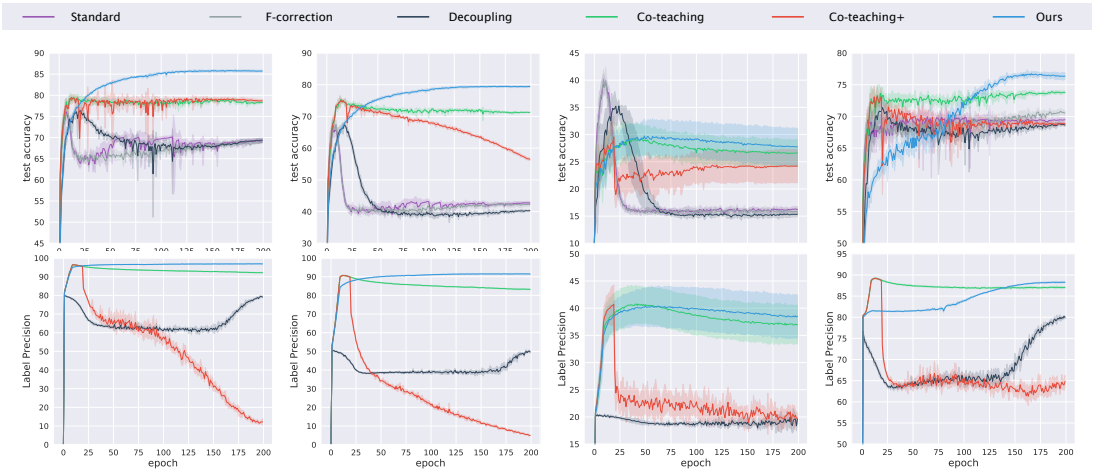
Flipping-Rate	Standard	F-correction	Decoupling	Co-teaching	Co-teaching+	JoCoR
Symmetry-20%	79.56 ± 0.44	95.38 ± 0.10	93.16 ± 0.11	95.10 ± 0.16	97.81 ± 0.03	$\mathbf{98.06} \pm 0.04$
Symmetry-50%	52.66 ± 0.43	92.74 ± 0.21	69.79 ± 0.52	89.82 ± 0.31	95.80 ± 0.09	$\mathbf{96.64} \pm 0.12$
Symmetry-80%	23.43 ± 0.31	72.96 ± 0.90	28.51 ± 0.65	79.73 ± 0.35	58.92 ± 14.73	$\mathbf{84.89} \pm 4.55$
Asymmetry-40%	79.00 ± 0.28	89.77 ± 0.96	81.84 ± 0.38	90.28 ± 0.27	93.28 ± 0.43	$\mathbf{95.24} \pm 0.10$

sample selection, so the algorithm with higher label precision is also more robust to the label noise. All experiments are repeated five times. The error bar for STD in each figure has been highlighted as a shade.

Selection setting. Following Co-teaching, we assume that the noise rate τ is known. To conduct a fair comparison in benchmark datasets, we set the ratio of small-loss samples $R(t)$ as identical: $R(t) = 1 - \min\left\{\frac{t}{T_k}\tau, \tau\right\}$, where $T_k = 10$ for MNIST, CIFAR-10 and CIFAR100, $T_k = 5$ for Clothing1M. If τ is not known in advance, τ can be inferred using validation sets [64, 161].

3.3.2 Comparison with the State-of-the-Arts

Results on *MNIST*. At the top of Figure 3.3, it shows test accuracy vs. epochs on *MNIST*. In all four plots, we can see the memorization effect of networks, i.e., test accuracy of Standard first reaches a very high level and then gradually decreases.



(A) S-20% (B) S-50% (C) S-80% (D) A-40%

FIGURE 3.4: Results on CIFAR-10 dataset. Top: test accuracy(%) vs. epochs; bottom: label precision(%) vs. epochs.

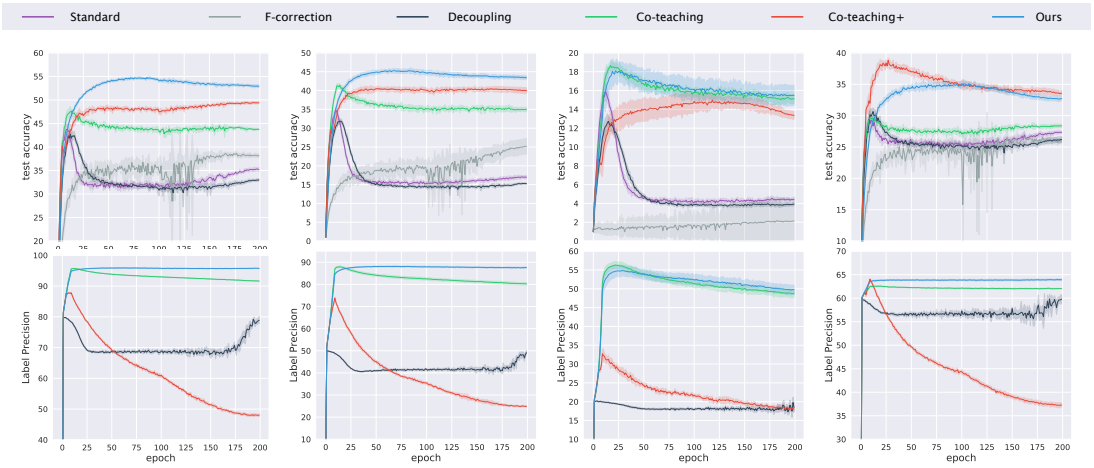
TABLE 3.5: Average test accuracy (%) on *CIFAR-10* over the last 10 epochs.

Flipping-Rate	Standard	F-correction	Decoupling	Co-teaching	Co-teaching+	JoCoR
Symmetry-20%	69.18 ± 0.52	68.74 ± 0.20	69.32 ± 0.40	78.23 ± 0.27	78.71 ± 0.34	**85.73** ± 0.19
Symmetry-50%	42.71 ± 0.42	42.19 ± 0.60	40.22 ± 0.30	71.30 ± 0.13	57.05 ± 0.54	**79.41** ± 0.25
Symmetry-80%	16.24 ± 0.39	15.88 ± 0.42	15.31 ± 0.43	26.58 ± 2.22	24.19 ± 2.74	**27.78** ± 3.06
Asymmetry-40%	69.43 ± 0.33	70.60 ± 0.40	68.72 ± 0.30	73.78 ± 0.22	68.84 ± 0.20	**76.36** ± 0.49

Thus, a good robust training method should stop or alleviate the decreasing process. On this point, JoCoR consistently achieves higher accuracy than all the other baselines in all four cases.

We can compare the test accuracy of different algorithms in detail in Table 3.4. In the most natural Symmetry-20% case, all new approaches work better than Standard obviously, which demonstrates their robustness. Among them, JoCoR and Co-teaching+ work significantly better than other methods. When it goes to Symmetry-50% case and Asymmetry-40% case, Decoupling begins to fail while other methods still work fine, especially JoCoR and Co-teaching+. However, Co-teaching+ cannot combat with the hardest Symmetry-80% case, where it only achieves 58.92%. In this case, JoCoR achieves the best average classification accuracy (84.89%) again.

To explain such excellent performance, we plot label precision vs. epochs at the bottom of Figure 3.3. Only Decoupling, Co-teaching, Co-teaching+ and JoCoR are considered here, as they include example selection during training. First, we can



(A) S-20% (B) S-50% (C) S-80% (D) A-40%

FIGURE 3.5: Results on CIFAR-100 dataset. Top: test accuracy(%) vs. epochs; bottom: label precision(%) vs. epochs.

TABLE 3.6: Average test accuracy (%) on *CIFAR-100* over the last 10 epochs.

Flipping-Rate	Standard	F-correction	Decoupling	Co-teaching	Co-teaching+	JoCoR
Symmetry-20%	35.14 ± 0.44	37.95 ± 0.10	33.10 ± 0.12	43.73 ± 0.16	49.27 ± 0.03	**53.01** ± 0.04
Symmetry-50%	16.97 ± 0.40	24.98 ± 1.82	15.25 ± 0.20	34.96 ± 0.50	40.04 ± 0.70	**43.49** ± 0.46
Symmetry-80%	4.41 ± 0.14	2.10 ± 2.23	3.89 ± 0.16	15.15 ± 0.46	13.44 ± 0.37	**15.49** ± 0.98
Asymmetry-40%	27.29 ± 0.25	25.94 ± 0.44	26.11 ± 0.39	28.35 ± 0.25	**33.62** ± 0.39	32.70 ± 0.35

see both JoCoR and Co-teaching can successfully pick clean instances out. Note that JoCoR not only reaches high label precision in all four cases but also performs better and better with the increase of epochs while Co-teaching declines gradually after reaching the top. This shows that our approach is better at finding clean instances. Then, Decoupling and Co-teaching+ fail in selecting clean examples. As mentioned in Related Work, very few examples are utilized by Co-teaching+ in the training process when noise rate goes to be extremely high. In this way, we can understand why Co-teaching+ performs poorly on the hardest case.

Results on *CIFAR-10*. Table 3.5 shows test accuracy on *CIFAR-10*. As we can see, JoCoR performs the best in all four cases again. In the Symmetric-20% case, JoCoR works much better than all other baselines and Co-teaching+ performs better than Co-teaching and Decoupling. In the other three cases, JoCoR is still the best and Co-teaching+ cannot even achieve comparable performance with Co-teaching.

Figure 3.4 shows test accuracy and label precision vs. epochs. JoCoR outperforms

TABLE 3.7: Classification accuracy (%) on the *Clothing1M* test set

Methods	Standard	F-correction	Decoupling	Co-teaching	Co-teaching+	JoCoR
best	67.22	68.93	68.48	69.21	59.32	**70.30**
last	64.68	65.36	67.32	68.51	58.79	**69.79**

all the other comparing approaches on both test accuracy and label precision. On label precision, while Decoupling and Co-teaching+ fail to find clean instances, both JoCoR and Co-teaching can do this. An interesting phenomenon is that in the Asymmetry-40% case, although Co-teaching can achieve better performance than JoCoR in the first 100 epochs, JoCoR consistently outperforms it in all the later epochs. The result shows that JoCoR has better generalization ability than Co-teaching.

Results on *CIFAR-100*. Then, we show our results on *CIFAR-100*. The test accuracy is shown in Table 3.6. Test accuracy and label precision vs. epochs are shown in Figure 3.5. Note that there are only 10 classes in *MNIST* and *CIFAR-10* datasets. Thus, overall the accuracy is much lower than previous ones in Tables 3.4 and 3.5. But JoCoR still achieves high test accuracy on this datasets. In the easiest Symmetry-20% and Symmetry-50% cases, JoCoR works significantly better than Co-teaching+, Co-teaching and other methods. In the hardest Symmetry-80% case, JoCoR and Co-teaching tie together but JoCoR still gets higher testing accuracy. When it turns to Asymmetry-40% case, JoCoR and Co-teaching+ perform much better than other methods. On label precision, JoCoR keeps the best performance in all four cases.

Results on *Clothing1M*. Finally, we demonstrate the efficacy of the proposed method on the real-world noisy labels using the *Clothing1M* dataset. As shown in Table 3.7, *best* denotes the scores of the epoch where the validation accuracy is optimal, and *last* denotes the scores at the end of training. The proposed JoCoR method gets better result than state-of-the-art methods on *best*. After all epochs, JoCoR achieves a significant improvement in accuracy of +5.11 over Standard, and an improvement of +1.28 over the best baseline method.

3.3.3 Ablation study

To conduct ablation study for analyzing the effect of Co-Regularization, we set up the experiments above *MNIST* and *CIFAR-10* with Symmetry-50% noise. For implementing Joint Training without Co-Regularization (*Joint-only*), we set the λ in (3.1) to 0. Besides, to verify the effect of the Joint Training paradigm, we introduce Co-teaching and Standard enhanced by "small-loss" selection (*Standard+*) to join the comparison. Recall that the joint-training method selects examples by the joint loss while Co-teaching uses cross-update method to reduce the error flow [27], these two methods should play a similar role during training according to the previous analysis.

The test accuracy and label precision vs. epochs on *MNIST* are shown in Figure 3.6. As we can see, JoCoR performs much better than the others on both test accuracy and label precision. The former keeps almost no decrease while the latter decline a lot after reaching the top. This observation indicates that Co-Regularization strongly hinders neural networks from memorizing noisy labels.

The test accuracy and label precision vs. epochs on *CIFAR-10* are shown in Figure Figure 3.7. In this figure, JoCoR still maintains a huge advantage over the other three methods on both test accuracy and label precision while Joint-only, Co-teaching and Standard+ remain the same trend as these for *MNIST*, keeping a downward tendency after increasing to the highest point. These results show that Co-Regularization plays a vital role in handling noisy labels. Moreover, Joint-only achieves a comparable performance with Co-teaching on test accuracy and performs better than Co-teaching and Standard+ on label precision. It shows that Joint Training is a more efficient paradigm to help select clean examples than cross-update in Co-teaching.

3.3.4 Parameter sensitivity analysis

To conduct the sensitivity analysis on parameter λ, we set up the experiments above *MNIST* with symmetry-50% noise. Specifically, we compare λ in the range of [0.05, 0.35, 0.65, 0.95]. The larger the λ is, the less the divergence of two classifiers in JoCoR would be.

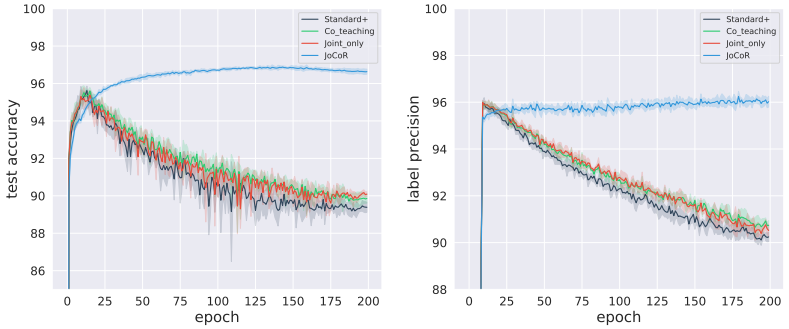


FIGURE 3.6: Results of ablation study on *MNIST*

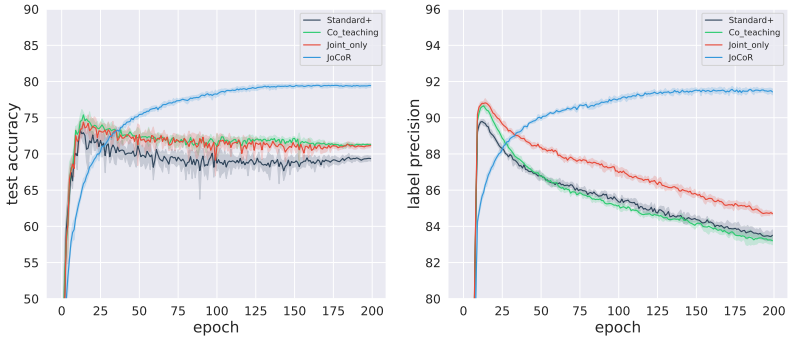
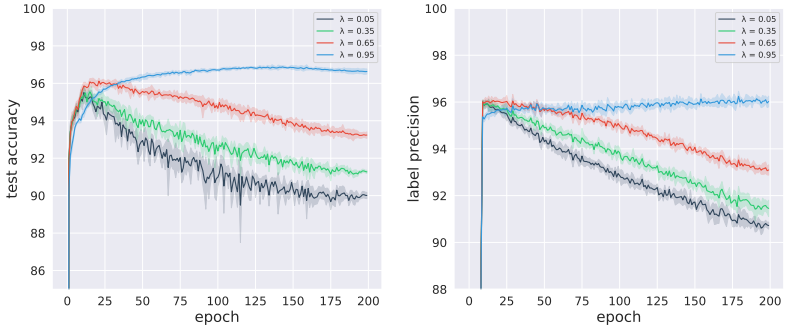


FIGURE 3.7: Results of ablation study on *CIFAR-10*



(A) test accuracy (B) label precision

FIGURE 3.8: Results of JoCoR with different λ on *MNIST*

The test accuracy and label precision vs. number of epochs are in Figure 3.8. Obviously, As the λ increases, the performance of our algorithm on test accuracy gets better and better. When $\lambda = 0.95$, JoCoR achieves the best performance. We can see the same trends on label precision, which means that JoCoR can select clean example more precisely with a larger λ.

3.4 Chapter Summary

In this chapter, we introduce an effective approach called JoCoR to improve the robustness of deep neural networks with noisy labels. The key idea of JoCoR is to train two classifiers simultaneously with one joint loss, which is composed of regular supervised part and Co-Regularized part. Similar to Co-teaching+, we also select small-loss instances to update networks in each mini-batch data by the joint loss. We conduct experiments on *MNIST*, *CIFAR-10*, *CIFAR-100* and *Clothing1M* to demonstrate that, JoCoR can train deep models robustly with the slightly and extremely noisy supervision. Furthermore, the ablation studies clearly demonstrate the effectiveness of Co-Regularization and Joint Training. In future work, we will explore the theoretical foundation of JoCoR based on the view of traditional Co-training algorithms [152, 153].

Chapter 4

Learning with Multiple Noisy Labels as a Union

4.1 Motivation

Deep Neural Networks (DNNs) have achieved remarkable success on various real-world applications over the past years, while they heavily rely on a large number of training examples with accurate labels. To alleviate this issue, crowdsourcing provides a potential solution for large-scale annotations, which aims to elicit the correct label from crowdsourced labels. However, the collected crowdsourced labels could be very noisy due to human mistakes, especially for some difficult tasks like image annotation [162–164] and music genre classification [165].

For training a classifier with noisy crowdsourced labels from multiple annotators, the key is how to abstract information from the imperfect crowdsourced labels. A traditional solution is to select the true label from crowdsourced labels by majority voting [166, 167] and train a classifier with the selected true label. However, this naive method would cause biased results because it simply assumes that all annotators are equally reliable, which is usually impractical since different annotators normally have different levels of expertise [52]. In order to capture the expertise levels of different annotators, some variants of Expectation-Maximization (EM) algorithm [52] were adapted to infer the ground-truth label and train the classifier alternately. Due to the iterative manner, EM-style algorithms are computationally expensive, especially when DNNs are used [163]. To improve the efficiency of

training with DNNs, recent studies (e.g., CrowdLayer [30] and SpeeLFC [53]) aim to train an end-to-end deep neural network with multiple parametric annotator-specific transition matrices. Although they have achieved satisfactory practical performance, they still suffer from *computational inefficiency* and do not hold *learning consistency*[1].

To further address the above two limitations, we propose a novel method named UnionNet, which is not only theoretically guaranteed but also experimentally effective and efficient. Specifically, unlike existing methods that either fit a given label from each annotator independently or fuse all the labels into a reliable one, we concatenate the one-hot encoded vectors of crowdsourced labels provided by all the annotators, which takes all the labeling information as a union and keeps it intact and coordinates multiple annotators. In this way, we can directly train an end-to-end deep neural network by maximizing the likelihood of the intact labeling information with only a transition matrix. In summary, the key contributions of this chapter are as follows.

- We propose UnionNet [2], a novel method that takes the labeling information provided by all annotators as a union, and trains an end-to-end DNN maximizing the likelihood of this union with only a parametric transition matrix.
- For theoretical guarantee, we prove the learning consistency of UnionNet by establishing an *estimation error bound*, which shows that obtained empirical risk minimizer by UnionNet would converge to the true risk minimizer as the amount of training data tends to infinity.
- For practical performance, we conduct extensive experiments on synthesized as well as real-world crowdsourced datasets, and the experimental results demonstrate that UnionNet is superior to the state-of-the-art counterparts.
- For training efficiency, we experimentally show that the training time of UnionNet is nearly the same as that of standard DNN trained with correct labels, which is significantly less than that of other end-to-end methods.

[1] Learning is consistent, if and only if the risk of the learned classifier converges to the risk of the optimal classifier, as the amount of training data approaches infinity, where the optimality is defined over a given hypothesis class.

[2] The work in this chapter has been published in [168].

4.2 Preliminaries

In this section, we introduce the problem setting of learning with crowdsourced labels and the formulation of CrowdLayer, which is a representative related work.

4.2.1 Problem setting

Let $\mathcal{D} = \{(\boldsymbol{x}_i, Y_i)\}_{i=1}^{n}$ be a crowdsourced dataset that includes n examples with k classes, where for each instance $\boldsymbol{x}_i \in \mathbb{R}^d$, we receive a set of crowdsourced labels $Y_i = \{\tilde{y}_i^j \mid j = 1 \dots m\}$, with $\tilde{y}_i^j$ representing the label provided by the j-th annotator in a set of m annotators. It is worth noting that the number of crowdsourced labels of different examples could be different. Hence $|Y_i|$ may not be equal to $|Y_j|$ if $i \neq j$. Following [23, 30, 52, 53], we also assume each instance $\boldsymbol{x}_i$ has its corresponding latent ground-truth label y_i and it is related to the crowdsourced labels Y_i. It is worth noting that learning with crowdsourced data is a multi-class classification problem, which means there is only a true label for each training instance. This setting is different from multi-label learning [54–58] where multiple true labels are provided for each each training instance. Under this setting, our goal is to train a classifier f based on a crowdsourced dataset $\mathcal{D} = \{(\boldsymbol{x}_i, Y_i)\}_{i=1}^{n}$ that could accurately predict the ground-truth label of an unseen instance in the test phase.

4.2.2 Formulation of CrowdLayer

Instead of using an alternating optimization manner of EM-style algorithms, CrowdLayer [30] aims to train an end-to-end DNN that simultaneously optimizes all the components. Specifically, CrowdLayer relies on a crowd layer that takes the softmax outputs of a standard DNN model as the input, and learns multiple parametric annotator-specific transition matrices to fit the given label of each annotator independently.

For each annotator j, we define an annotator-specific crowd layer by a transition matrix $\boldsymbol{T}^j$, which is equivalent to a linear layer without bias, i.e., $\boldsymbol{T}^j\widehat{\boldsymbol{y}}$. In CrowdLayer, a transition matrix is explicitly learned for each annotator, thereby enabling

crowdsourced labels to propagate errors through the whole network structure. Formally, let $\widehat{\boldsymbol{y}}$ be the softmax output of a standard classifier. Given a loss function $\mathcal{L}$ (e.g., categorical cross-entropy loss) for classification, CrowdLayer minimizes the following objective:

$$\min_{\boldsymbol{\Theta}} \frac{1}{n} \sum\nolimits_{i=1}^{n} \sum\nolimits_{j=1}^{m} \mathcal{L}\big(\text{softmax}(\boldsymbol{T}^j \widehat{\boldsymbol{y}}_i), \tilde{y}_i^j\big), \tag{4.1}$$

where $\boldsymbol{\Theta} = \{\{\boldsymbol{T}^j\}_{j=1}^m, f\}$ denotes the set of all learning parameters that include the parameters of transition matrices and the classifier. Trained with the above objective function, CrowdLayer achieves the goal of directly learning from crowdsourced labels in an end-to-end manner. However, CrowdLayer does not hold learning consistency. In addition, most existing end-to-end methods, including CrowdLayer and SpeeLFC, need to forward propagate through multiple linear layers in a for-loop way. It also introduces extra computational complexity, especially when scaling to a large number of annotators.

4.3 The Proposed Approach

As described above, the training inefficiency of existing end-to-end algorithms is caused by the for-loop forward propagation of multiple independent annotator-specific transition matrices. Hence we conjecture that if we use a single transition matrix, the training efficiency may be improved.

4.3.1 Union of crowdsourced labels

We present a novel end-to-end method called UnionNet, which allows us to leverage a single transition matrix to directly train a DNN with the crowdsourced labels provided by multiple annotators. UnionNet not only experimentally outperforms other end-to-end counterparts, but also holds learning consistency and has high training efficiency. As shown in Figure 1(b), instead of learning an independent annotator-specific transition matrix for each annotator, we propose to learn a single transition matrix. In this way, we could transform the softmax outputs of the standard classifier via this transition matrix to fit a union of labeling information provided by all the annotators.

By taking the labeling information provided by all the annotators as a union, our proposed UnionNet is advantageous in two aspects. First, unlike previous methods [30, 53] that treat each annotator equally, UnionNet naturally coordinates the contributions of different annotators on the true label. Specifically, in CrowdLayer, each annotator would contribute a loss independently, while in our UnionNet, there is only a single loss that is contributed by all the annotators. Therefore, UnionNet takes into account the collaboration of all the annotators, while previous methods treat each annotator independently. Hence UnionNet is expected to achieve better performance, especially in the correlated settings. Second, previous methods conduct forward propagation through multiple linear layers (i.e., transition matrices) in a for-loop way. In contrast, UnionNet only has a single transition matrix, hence it avoids that for-loop way. Therefore, UnionNet would be more computationally efficient.

Therefore, the question becomes how to combine the crowdsourced labels of all the annotators as a union. Our proposed UnionNet provides a simple yet effective answer to this question. Concretely, we can concatenate the vectors of one-hot encoded crowdsourced labels provided by different annotators to form a united vector $\widetilde{\boldsymbol{y}}_i$:

$$\widetilde{\boldsymbol{y}}_i = \text{concatenate}(\boldsymbol{e}^{(\tilde{y}_i^1)}, \boldsymbol{e}^{(\tilde{y}_i^2)}, \ldots, \boldsymbol{e}^{(\tilde{y}_i^m)}). \tag{4.2}$$

where $\boldsymbol{e}^{(\tilde{y}_i^j)} := (0, \ldots, 1, \ldots, 0)^\top \in \{0,1\}^k$ is a one-hot vector and only the $\tilde{y}_i^j$-th entry of $\boldsymbol{e}^{(\tilde{y}_i^j)}$ is 1. When the j-th annotator does not provide a label for the i-th instance $\boldsymbol{x}_i$, we define the vector $\boldsymbol{e}^{(\tilde{y}_i^j)}$ as a zero vector. In this way, all the crowdsourced labels are processed as a union. Here, one may think that we could use another combination method to produce a union of crowdsourced labels. For example, we could sum up the vectors of one-hot encoded crowdsourced labels together to create a single union vector. However, this simple addition operation could not fully capture the complex relationships between the crowdsourced labels and the ground-truth label. In contrast, our proposed concatenation operation in Eq. (4.2) enables the union of crowdsourced labels to contain the information of annotators so that it can automatically coordinate the influences of different annotators. We also experimentally demonstrate the advantage of our concatenation operation over the addition operation by conducting ablation studies in the experiment section.

4.3.2 Training objective

By taking the crowdsourced labels as a union vector $\tilde{\boldsymbol{y}}_i$, our final goal is to train an end-to-end deep model by fitting this union vector $\tilde{\boldsymbol{y}}_i$. To achieve this goal, we adopt the widely used *maximum likelihood principle* [169, 170]. Formally speaking, we would like to maximize the $p(\widetilde{\boldsymbol{y}} \mid \boldsymbol{x})$, where $p(\widetilde{\boldsymbol{y}} \mid \boldsymbol{x})$ can be formulated as

$$\begin{aligned} p(\widetilde{\boldsymbol{y}} \mid \boldsymbol{x}) = \sum\nolimits_y p(\widetilde{\boldsymbol{y}}, y \mid \boldsymbol{x}) &= \sum\nolimits_y p(\widetilde{\boldsymbol{y}} \mid y, \boldsymbol{x}) p(y \mid \boldsymbol{x}) \\ &= \sum\nolimits_y p(\widetilde{\boldsymbol{y}} \mid y) p(y \mid \boldsymbol{x}), \end{aligned} \tag{4.3}$$

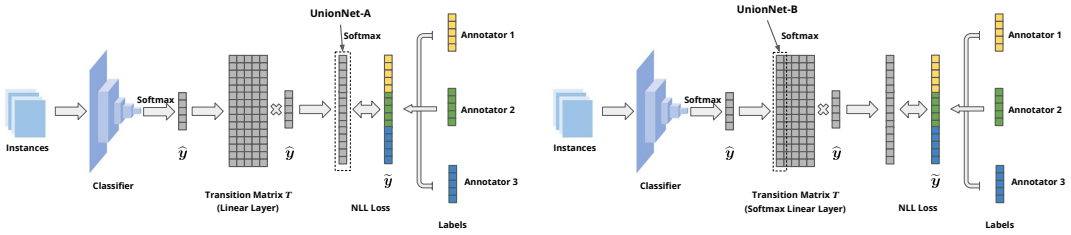
where the last equality holds due to the widely used assumption [30, 52, 53] that the crowdsourced labels are only related to the ground-truth label y and independent of the instance $\boldsymbol{x}$. It is worth noting that there is a similar assumption in the area of noisy-label learning [27, 59, 67, 171–181] that the observed label noise is class-conditional. In this way, the expected risk of our proposed UnionNet could be formulated as the *negative expected log-likelihood*:

$$R(f) = -\mathbb{E}_{p(\boldsymbol{x}, \widetilde{\boldsymbol{y}})} \big[\log p(\widetilde{\boldsymbol{y}} \mid \boldsymbol{x}) \big], \tag{4.4}$$

where $p(\widetilde{\boldsymbol{y}} \mid \boldsymbol{x})$ denotes the distribution of crowdsourced data. In terms of Eq. (4.3), the key of our proposed end-to-end learning method is how to empirically approximate the union vector $\widetilde{\boldsymbol{y}}_i$ by using different components in our deep archtecture. Here, we provide two solutions, which are named UnionNet-A and UnionNet-B respectively.

UnionNet-A. Motivated by the training objective of CrowdLayer [30] in Eq. (4.1), we could empirically approximate the union vector $\widetilde{\boldsymbol{y}}_i$ of $\boldsymbol{x}_i$ by $\text{softmax}(\boldsymbol{T}\widehat{\boldsymbol{y}}_i)$ where $\widehat{\boldsymbol{y}}_i$ denotes the softmax outputs of the standard classifier and $\boldsymbol{T} \in \mathbb{R}^{km} \times \mathbb{R}^k$ is a parametric transition matrix, which is defined as a linear layer without bias. It is worth noting that we need to further conduct the softmax operation on $\boldsymbol{T}\widehat{\boldsymbol{y}}_i$ to make it always non-negative, as there is no constraints posed on the transition matrix $\boldsymbol{T}$. Guided by the expected risk in Eq. (4.4), we have the following training objective of UnionNet-A:

$$\min_{\boldsymbol{\Theta}} -\frac{1}{n} \sum\nolimits_{i=1}^{n} \widetilde{\boldsymbol{y}}_i \log \big(\text{softmax}(\boldsymbol{T}\widehat{\boldsymbol{y}}_i) \big), \tag{4.5}$$



(A) UnionNet-A (B) UnionNet-B

FIGURE 4.1: The training schemes of UnionNet (ours) for the classification task with 5 classes and 3 annotators. For the two algorithms of UnionNet, softmax operation is either applied on transformed outputs for UnionNet-A or columns of the transition matrix for UnionNet-B. In the test stage, $\widehat{\boldsymbol{y}}$ can be readily used to make predictions for an unseen instance $\boldsymbol{x}_i$.

where $\boldsymbol{\Theta} = \{\boldsymbol{T}, f\}$ denotes the set of all learning parameters. It is worth noting that for UnionNet-A, the parametric transition matrix $\boldsymbol{T}$ is not able to model the conditional probability $p(\widetilde{\boldsymbol{y}} \mid y)$ since the elements of $\boldsymbol{T}$ may be smaller than 0. As a consequence, UnionNet-A may not hold any theoretical guarantee, since it does strictly follow from Eq. (4.3) and Eq. (4.4). To alleviate this issue, we further present the other solution UnionNet-B, which is theoretically guaranteed.

UnionNet-B. Motivated by SpeeLFC [53] that employs a probabilistic parametric transition matrix to enhance the interpretability for each annotator, we could also empirically approximate the union vector $\widetilde{\boldsymbol{y}}_i$ of $\boldsymbol{x}_i$ by $\text{softmax}(\boldsymbol{T})\widehat{\boldsymbol{y}}_i$ where $\text{softmax}(\boldsymbol{T})$ means taking the softmax operation on each column of the parametric transition matrix $\boldsymbol{T}$, which is a linear layer without bias. It is worth noting that each column of $\text{softmax}(\boldsymbol{T})$ indicates that the probability vector of each true label y becomes a specific union vector $\widetilde{\boldsymbol{y}}$. Therefore, we need to impose the softmax operation on each column of $\boldsymbol{T}$ to satisfy the requirements of $p(\widetilde{\boldsymbol{y}} \mid y)$, i.e., $\sum_{\widetilde{\boldsymbol{y}}} p(\widetilde{\boldsymbol{y}} \mid y) = 1$ and $p(\widetilde{\boldsymbol{y}} \mid y) \geq 0$ for any $\widetilde{\boldsymbol{y}}$ and y. Other weakly supervised learning settings have also provided various attempts to estimate such kind of probabilistic transition matrix using *anchor points* [29, 67], or learn it as a noise adaptation layer [156]. Our proposed UnionNet-B adopts the latter strategy, and the training objective of UnionNet-B is expressed as

$$\min_{\boldsymbol{\Theta}} -\frac{1}{n}\sum\nolimits_{i=1}^{n} \widetilde{\boldsymbol{y}}_i \log\left(\text{softmax}(\boldsymbol{T})\widehat{\boldsymbol{y}}_i\right), \tag{4.6}$$

where $\boldsymbol{\Theta} = \{\boldsymbol{T}, f\}$ denotes the set of all learning parameters. It is worth noting

Algorithm 2: UnionNet

Input: Classifier f with $\boldsymbol{W}$, transition matrix $\boldsymbol{T}$, number of epochs $E_{\max}$;

1: **for** $e = 1,2,\ldots,E_{\max}$ **do**
2: $\quad\widehat{\boldsymbol{y}} = f(\boldsymbol{x}, \boldsymbol{W})$;
3: $\quad$**Obtain** union $\widetilde{\mathbf{y}}$ by Eq. (2);
4: $\quad$**Obtain** $\mathcal{L}$ by Eq. (5) for UnionNet-A or Eq. (6) for UnionNet-B based on $\widetilde{\boldsymbol{y}}$ and $\widehat{\boldsymbol{y}}$;
5: $\quad$**Update** $\{\boldsymbol{W}, \boldsymbol{T}\}$ by $\mathcal{L}$ using Adam optimization method.
6: **end for**

Output: $\boldsymbol{W}$

that SpeeLFC employs multiple probabilistic parametric transition matrices, each for an annotator, and it does not hold any guarantee. In contrast, our proposed UnionNet-B only uses a single probabilistic parametric transition matrix, and we will show that it is theoretically consistent and has higher training efficiency.

Differences between UnionNet-A and UnionNet-B. As shown in Figure 4.1, both the two proposed methods optimize a single probabilistic parametric transition matrix. Differently, UnionNet-A applies the softmax operation on transformed outputs while UnionNet-B uses it on columns of the transition matrix. As a result, UnionNet-B is theoretically consistent but UnionNet-A do not hold any theoretical guarantee. In addition, the softmax operation on columns of the parametric transition matrix provides a better way to automatically coordinate the influences of different annotators, which is clearly supported by the empirical results in Table 4.2. With such a guarantee, UnionNet-B is expected to achieve better empirical performance than UnionNet-A.

4.3.3 Initialization strategy

A good initialization strategy for the parametric transition matrix $\boldsymbol{T}$ is helpful for the convergence of the whole network architecture in the training stage. At the beginning of training, we could treat the linear layer (which represents the whole transition matrix) as a simple concatenation of m sub-layers and each of them represents a transition matrix of a annotator: $\boldsymbol{T} = \text{concatenate}(\boldsymbol{T}^1, \boldsymbol{T}^2, \cdots, \boldsymbol{T}^m)$. We adopt the following two initialization strategies in the experiments.

For the first initialization strategy, we follow CrowdLayer [30] to initialize the sub-layer for each annotator as

$$\boldsymbol{T}^r(a,b) = (1-\epsilon)\mathbb{I}_{\{a=b\}} + \frac{\epsilon}{C-1}\mathbb{I}_{\{a\neq b\}}, \tag{4.7}$$

where $\mathbb{I}_{condition} = 1$ when *condition* is *true*, $\mathbb{I}_{condition} = 0$ when *condition* is *false*, and ϵ denotes an extremely small constant, which is fixed at 10^{-5} in our experiments.

For the second initialization strategy, we follow [52, 182] to initialize the sub-layer for each annotator as

$$\boldsymbol{T}^r(a,b) = \log \frac{\sum_{i=1}^{n} Q\left(y_i = a\right) \mathbb{I}_{\{\tilde{y}_i^j = b\}}}{\sum_{i=1}^{n} Q\left(y_i = a\right)}, \tag{4.8}$$

where $Q\left(y_i = a\right) := \frac{1}{m}\sum_{r=1}^{m} \mathbb{I}_{\{\tilde{y}_i^j = a\}}$.

We will provide detailed information about which strategy will be adopted on which dataset in the experiments. The details of our proposed UnionNet are shown in Algorithm 2. Specifically, UnionNet is more computationally efficient since it only optimizes a single transition matrix during training, thereby avoiding the for-loop way in previous methods. Moreover, UnionNet explores the relationships between annotators by the concatenate operation, thereby being able to achieve better practical performance.

4.4 Theoretical Analysis

For UnionNet, the crowdsourced labels for each instance are taken as a union vector. In this way, we could represent the provided crowdsourced datasets as $\{\boldsymbol{x}_i, \widetilde{\boldsymbol{y}}_i\}_{i=1}^n$, where each example is assumed to be independently sampled from the distribution of crowdsourced data $p(\boldsymbol{x}, \widetilde{\boldsymbol{y}})$. It is noteworthy that for UnionNet-B, softmax($\boldsymbol{T}$) could perfectly model the conditional probability $p(\widetilde{\boldsymbol{y}} \mid y)$ in Eq. (4.3). Hence we only focus on the theoretical analysis of UnionNet-B, and represent the used transition matrix softmax($\boldsymbol{T}$) as $\widetilde{\boldsymbol{T}}$.

We define the model class as $\mathcal{F} \subset \{f : \mathbb{R}^d \mapsto \mathbb{R}^k\}$ and define the true risk minimizer as

$$f^\star = \arg\min_{f\in\mathcal{F}} \mathbb{E}_{p(\boldsymbol{x},y)}\big[\mathcal{L}(f(\boldsymbol{x}), y)\big], \tag{4.9}$$

where $p(\boldsymbol{x}, y)$ denotes the underlying data distribution of the normal example $(\boldsymbol{x}, y)$ and y is the true label of $\boldsymbol{x}$. Then we define the obtained empirical risk minimizer by minimizing Eq. (4.6) as $\widehat{f}$. In the following, we will prove that the obtained empirical risk minimizer $\widehat{f}$ would converge to the true risk miminizer $f^\star$ as the collected crowdsourced training data approaches infinity.

First of all, we prove that the obtained expected risk minimizer $\widetilde{f}^\star$ by minimizing Eq. (4.4) is exactly the same as the true risk minimizer $f^\star$ if the categorical cross entropy loss is used in Eq. (4.9). All the proofs are provided in Appendix A.

Lemma 4.1. *Suppose $g(\boldsymbol{x}) = \mathrm{softmax}(f(\boldsymbol{x}))$, by optimizing Eq. (4.9) with categorical cross entropy loss, the optimal mapping $g^\star$ (w.r.t. $f^\star$) satisfies $g_i^\star(\boldsymbol{x}) = p(y = i \mid x), \forall i \in [k]$.*

Theorem 4.2. *When the transition matrix $\widetilde{\boldsymbol{T}}$ has full rank and the condition in Lemma 4.1 is satisfied, our expected risk minimizer $\widetilde{f}^\star$ is the same as the true risk minimizer $f^\star$, i.e., $\widetilde{f}^\star = f^\star$.*

Next, we theoretically establish an estimation error bound, which demonstrates the learning consistency of UnionNet-B based on *Rademacher complexity* [183]. From Eq. (4.6), we can introduce a composite loss $\widetilde{\mathcal{L}}(f(\boldsymbol{x}), \widetilde{\boldsymbol{y}}) = -\widetilde{\boldsymbol{y}} \log(\widetilde{\boldsymbol{T}}\mathrm{softmax}(f(\boldsymbol{x})))$. Let $\mathcal{F}$ be represented as $\mathcal{F} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_k$, where $\mathcal{F}_i := \{f : \boldsymbol{x} \mapsto f_i(\boldsymbol{x}) \mid f \in \mathcal{F}\}, \forall i \in [k]$. We denote by $\mathfrak{R}_n(\mathcal{F}_y)$ the Rademacher complexity of $\mathcal{F}_y$ given the sample size n over $p(\boldsymbol{x})$. Then we have the following theorem.

Theorem 4.3. *Assume $\widetilde{\mathcal{L}}$ is ρ-Lipschitz ($\rho < 0 < \infty$) continuous w.r.t. $f(\boldsymbol{x})$ and upper bounded by M, i.e., for any $(\boldsymbol{x}, \widetilde{\boldsymbol{y}}) \sim p(\boldsymbol{x}, \widetilde{\boldsymbol{y}})$, $\widetilde{\mathcal{L}}(f(\boldsymbol{x}), \widetilde{\boldsymbol{y}}) \leq M$. Then, for any $\delta > 0$, with probability at least $1 - \delta$,*

$$R(\widehat{f}) - R(f^\star) \leq 2\sqrt{2}\rho \sum\nolimits_{y=1}^{k} \mathfrak{R}_n(\mathcal{F}_y) + M\sqrt{\frac{\log\frac{2}{\delta}}{2n}}. \tag{4.10}$$

Generally, $\mathfrak{R}_n(\mathcal{F}_y)$ can be bounded by $C_{\mathcal{F}}/\sqrt{n}$ for a positive constant $C_{\mathcal{F}}$ [173, 184, 185]. Hence Theorem 4.3 demonstrates that the empirical risk minimizer $\widehat{f}$

(obtained by learning from only crowdsourcing data) converges to the true risk minimizer $f^\star$ (obtained by minimizing the expected classification risk on fully laebled data) as the number of crowdsourcing data approaches infinity ($n \to \infty$). Such a theoretical result indicates that we can obtain a reasonable classifier by directly using our UnionNet-B with only crowdsourcing data, and such a classifier would be better if more crowdsourcing data are provided. In addition, we can observe that the convergence rate of the derived bound in Theorem 2 is $\mathcal{O}_p(1/\sqrt{n})$ where $\mathcal{O}_p$ denotes the order in probability. This order is known as the optimal parametric rate for empirical risk minimization without additional assumptions [186].

4.5 Experiments

4.5.1 Benchmark datasets

Synthesized datasets. Four widely used synthesized datasets are used in our experiments [30, 182, 187]. **Dogs vs. Cats** dataset consists of 25000 images from 2 classes dogs and cats, which is split into a 12500-image training set and a 12500-image test set. **CIFAR-10** dataset consists of 60000 images from 10 classes, which is split into a 50000-image training set and a 10000-image test set. **MNIST** dataset also contains 60,000 training images and 10,000 testing images. **LUNA16** dataset consists of 888 CT scans for lung nodule. We preprocessed the CT scans by generating 8106 50×50 gray-scale images, which is split into a 6484-image training set and a 1622-image testing set.

For each synthesized dataset, we follow [182] to generate two groups of crowdsourced labels: labels provided by (H) annotators with relatively high expertise; (L) annotators with relatively low expertise. Besides, we consider two cases of mistakes: **independent mistakes**, where senior annotators are mutually conditionally independent, and **correlated mistakes**, where senior annotators are mutually conditional independent, and each junior annotators copies one of the senior annotators. For each of the situations (H) and (L), the two cases have the same senior annotators. For independent mistakes, we set the number of annotators to 5 and 10 for the situations (H) and (L), respectively. For correlated mistakes, the number of senior annotators is set to 5 and 10 for the situations (H) and (L), while

the number of junior annotators is set to 5 and 2 for the situations (H) and (L). In the following, we will introduce the details of the two cases for different datasets.

Independent mistakes. For Dogs vs. Cats, in situation (H), some senior annotators are more familiar with cats, while others make better judgments on dogs. Specifically, the expertise of five annotators are A: 0.6/0.8, B: 0.6/0.6, C: 0.9/0.6, D: 0.7/0.7, E: 0.6/0.7. For example, the expertise of annotator A is 0.6/0.8 in the sense that if the ground truth is dog/cat, she labels the image as "dog"/"cat" with probability 0.6/0.8 respectively. In situation (L), all ten seniors' expertise are 0.55/0.55.

For CIFAR-10, in situation (H), we generate annotators who may make mistakes in distinguishing the hard pairs: cat/dog, deer/horse, airplane/bird, automobile/trunk, frog/ship, but can perfectly distinguish other easy pairs (e.g. cat/frog), which makes sense in practice. When they cannot distinguish the pair, some of them may label the pair randomly and some of them label the pair the same class. In detail, for each hard pair, annotator A label the pair the same class (e.g. A always labels the image as "cat" when the image has cats or dogs), annotator B labels the pair uniformly at random (e.g. B labels the image as "cat" with the probability 0.5 and "dog" with the probability 0.5 when the image has cats or dogs). Annotator C is familiar with mammals so she can distinguish cat/dog and deer/hose, while for other hard pairs, she label each of them uniformly at random. Annotator D is familiar with vehicles so she can distinguish airplane/bird, automobile/trunk and frog/ship, while for other hard pairs, she always label each of them the same class. Annotator E does not have special expertise. For each hard pair, annotator E labels them correctly with the probability 0.6. In situation (L), all ten senior experts label each image correctly with probability 0.2 and label each image as other false classes uniformly with probability $\frac{0.8}{9}$. We use the same settings for the MNIST dataset.

For LUNA16, in situation (H), some senior annotators tend to label the image as "benign" while others tend to label the image as "malignant". Their expertise for benign/malignant are: A: 0.6/0.9, B: 0.7/0.7, C: 0.9/0.6, D: 0.6/0.7, E: 0.7/0.6. In situation (L), all ten seniors' expertise are 0.6/0.6.

Correlated mistakes. For Dogs vs. Cats, MNIST, CIFAR-10 and LUNA16, in situation (H), two junior annotators copy annotator A's labels and three junior

annotators copy annotator C's labels; in situation (L), one junior annotator copies annotator A's labels and another junior annotator copies annotator C's labels.

Real-world datasets. Two widely used real-world datasets are used in our experiments [30, 53]. **LabelMe** dataset consists of a total of 2688 images, where 1000 of them are used to obtain labels from multiple annotators from Amazon Mechanical Turk [188] and the remaining 1688 images are used for evaluation. Each image is labeled by an average of 2.547 workers, with a mean accuracy of 69.2% [189]. **Music Genre Classification (MGC)** dataset contains 700 examples (with crowdsourced annotations) of songs with 30 seconds in length and are divided into 10 different music genres (classical, country, disco, etc.) [165].

Initialization. For the transition matrix in our algorithms, we use Eq. (4.8) for initialization on CIFAR10 and MGC. On other datasets, we initialize the layer by Eq. (4.7). The same setting is applied for CrowdLayer. For SpeeLFC [53], we use the setting described in their paper: the values on the diagonal elements of each transition matrix are initially set to 4.7, and the other values are fixed at 1.

4.5.2 Compared algorithms

We compare the two algorithms of UnionNet with the following state-of-the-art algorithms: **MajorVote** [166], which trains the network with the majority voting labels from all the annotators. **CrowdLayer** [30], which directly learns from multiple annotators' labels with multiple annotator-specific linear layers. **DoctorNet** [162] which models multiple annotators individually with different softmax output layers in the deep architecture. In this baseline, all the annotators are equally weighted. **WDN** [162], which is a variant of DocterNet with learnable weights for different annotators. **SpeeLFC** [53], which is a novel end-to-end probabilistic model that learns interpretable parameters of the crowd layer.

We conduct all the experiments on NVIDIA RTX 2080Ti GPUs. Specifically, we repeat the experiments 5 times with different random seeds and calculate the average test accuracy and standard deviation for reporting the results.

4.5.3 Training settings

Network Structure. We use the same base model as the classifier for all algorithms. Following [30], we employ the 4-layer CNN network as the classifier on Dogs vs. Cats and LUNA16. We use VGG-16 as the classifier on CIFAR10 and use 2-layer MLP on MNIST. On LabelMe, we also follow [30] to use the pre-trained VGG-16 model and apply an FC layer (with 128 units and ReLU activation) and an output layer on top with 50% dropout. On MGC, we use a linear model following the setting of SpeeLFC [53].

Optimizer. We use the Adam optimizer [190] for all the experiments. The hyperparameters like learning rate for different datasets are provided in Table 4.1.

TABLE 4.1: Summary of training setting for different datasets.

	Batch Size	Learning rate	Training epochs
Dogs vs. Cats	16	10^{-4}	20
CIFAR-10	64	10^{-3}	50
LUNA16	16	10^{-4}	20
LabelMe	64	10^{-4}	20
MGC	100	10^{-3}	5000

4.5.4 Experimental results

Synthesized Datasets. Table 4.2 shows the test accuracy of each algorithm on the four synthesized datasets under different crowdsourced settings. As we can see, our proposed algorithms achieve the best performance in most cases. On MNIST, UnionNet-B is the only algorithm that keeps the best or comparable performance across all the four cases, while CrowdLayer and UnionNet-A perform bad in the correlated (H) case. Surprisingly, we observed an abnormal phenomenon that almost all algorithms except UnionNet-B provide worse results on two high-expertise cases than on the low-expertise cases. Recall that we assume that the annotators' expertise in situation (H) may vary across different classes. Consequently, the results we obtained seem to indicate that on multi-class datasets learning with noisy high-expertise annotations that needs to take into account class-level differences in expertise might be even more challenging than learning with simple low-expertise annotations(i.e., high noise rate).

TABLE 4.2: Average test accuracy (%) with standard deviation on synthesized datasets under different settings (over 5 trials). The best results are highlighted in bold.

Datasets	Settings	Baselines					Ours	
		MajorVote	CrowdLayer	DoctorNet	WDN	SpeeLFC	UnionNet-A	UnionNet-B
MNIST	Independent(H)	74.98±1.47	98.29±0.07	75.63±0.70	76.71±0.65	50.05±0.03	94.37±2.22	**98.39±0.05**
	Correlated(H)	83.43±1.28	78.58±11.15	74.92±0.75	56.10±0.80	50.03±0.04	79.57±1.83	**98.37±0.06**
	Independent(L)	89.94±1.01	95.58±0.15	93.97±0.25	93.77±0.97	95.51±0.90	95.54±0.16	**95.62±0.15**
	Correlated(L)	91.86±0.54	95.45±0.23	93.80±0.29	93.34±0.46	95.31±0.36	**95.55±0.22**	95.38±0.20
CIFAR10	Independent(H)	68.25±0.79	56.06±6.03	69.98±0.23	45.42±0.25	40.21±3.61	52.60±1.27	**78.40±3.35**
	Correlated(H)	67.81±2.67	68.31±5.62	67.77±1.90	45.32±0.11	35.39±0.17	45.10±0.62	**76.99±0.43**
	Independent(L)	60.95±0.82	68.21±1.11	66.07±1.41	46.52±4.97	66.37±1.46	67.49±0.42	**68.66±0.85**
	Correlated(L)	60.10±0.85	65.30±0.96	63.01±3.83	39.17±3.46	64.63±1.26	63.20±1.38	**65.82±1.06**
Dog vs. Cat	Independent(H)	77.62±0.37	**78.84±0.48**	76.87±0.2	77.13±0.75	78.61±0.62	78.60±0.51	78.66±0.83
	Correlated(H)	77.63±0.55	78.64±0.71	77.07±0.43	75.26±1.12	78.65±0.60	**78.79±0.66**	78.65±0.50
	Independent(L)	60.21±0.76	**70.05±0.58**	65.77±1.44	65.45±0.67	69.69±1.08	69.71±0.44	69.74±0.37
	Correlated(L)	61.25±1.11	68.44±1.17	65.49±1.07	64.73±0.95	68.58±1.23	68.54±0.56	**68.59±1.17**
LUNA16	Independent(H)	90.59±0.43	91.10±0.36	89.66±0.33	89.76±0.42	91.27±0.27	**91.29±0.39**	91.03±0.12
	Correlated(H)	90.45±0.22	91.00±0.18	89.84±0.30	89.65±0.33	91.05±0.40	90.95±0.26	**91.13±0.19**
	Independent(L)	88.05±0.89	89.10±0.55	87.90±0.73	88.01±0.71	**89.53±0.55**	89.44±0.41	89.40±0.39
	Correlated(L)	87.52±1.08	89.03±0.37	87.72±0.66	87.88±0.82	88.99±0.53	88.95±0.32	**89.17±0.51**

On CIFAR10, UnionNet-B significantly outperforms all baselines in all the four settings, especially for experts with relatively high expertise. Specifically, UnionNet-B achieves significant improvement by about 12% over the second-best algorithms across the two cases with high expertise. In the two cases with low expertise, we observe that UnionNet-B still achieves the best performance while MajorVote performs much worse than the other baselines. It is worth noting that SpeeLFC performs badly on the two cases with relatively high expertise of both the MNIST dataset and CIFAR10 dataset, which shows the poor applicability of SpeeLFC to different cases.

On Dog vs. Cat and LUNA16 datasets, we observe that all the methods exhibit similarly high accuracy in the two cases with high expertise. A possible reason is that there are only 2 classes for this dataset. An interesting phenomenon is that our proposed algorithms consistently outperform all the baselines under all the settings with correlated mistakes, while they are slightly inferior but still comparable to the best baseline in three of the settings with independent mistakes. It shows that our proposed method has an advantage in handling correlated mistakes, which are closer to real-world settings. This phenomenon can be interpreted by that our UnionNet can naturally coordinate the contributions of different annotators on the true label and explores the relationships between the annotators, while previous methods treat each annotator independently. Therefore, our UnionNet is better suited for correlated mistakes than independent mistakes. When it comes to

TABLE 4.3: Average test accuracy (%) and standard deviation (over 5 trials) on real-world datasets.

Method	LabelMe	MGC
MajorVote	79.81±0.12	62.93±0.13
CrowdLayer	83.94±0.25	69.00±0.63
DoctorNet	79.97±0.40	57.07±0.25
WDN	81.08±0.22	41.47±0.50
SpeeLFC	83.70±0.33	48.40±3.22
UnionNet-A	83.92±0.21	68.60±0.53
UnionNet-B	**84.18±0.12**	**69.33±0.58**

the settings with independent mistakes, our UnionNet would achieve comparable performance to CrowdLayer since there is no relationship between the annotators.

Real-World Datasets. We also demonstrate the efficacy of the proposed methods on the real-world crowdsource datasets using LabelMe and MGC datasets in Table 4.3. On LabelMe Dataset, UnionNet-B gets the best result, while the performance of UnionNet-A is comparable to CrowdLayer and SpeeLFC. On MGC dataset, UnionNet-B still performs the best among all the method and UnionNet-A obtains comparable result with CrowdsLayer, while SpeeLFC is stuck at local minimum with poor performance.

To demonstrate the ability of our UnionNet to learn the reliabilities of the annotators, we compare the learned weight matrices of UnionNet-A and UnionNet-B with those of CrowdLayer, SpeeLFC, and the corresponding real confusion matrix on LabelMe dataset. Following [30], we select four annotators with the largest number of annotations. The results are shown in Figure 4.2. The real transition matrices of annotators are calculated based on their annotations and the ground truth, while the learned weight matrices of every algorithm are normalized for fair comparisons. From Figure 4.2, we observe that the learned matrices of UnionNet-A and UnionNet-B are closer to the real confusion matrix than those of CrowdLayer and SpeeLFC. This observation validates the effectiveness of our UnionNet, as UnionNet could automatically coordinate the influences of different annotators on the true label, while other algorithms only focus on each annotator independently without explicit consideration of the relationships among the annotators.

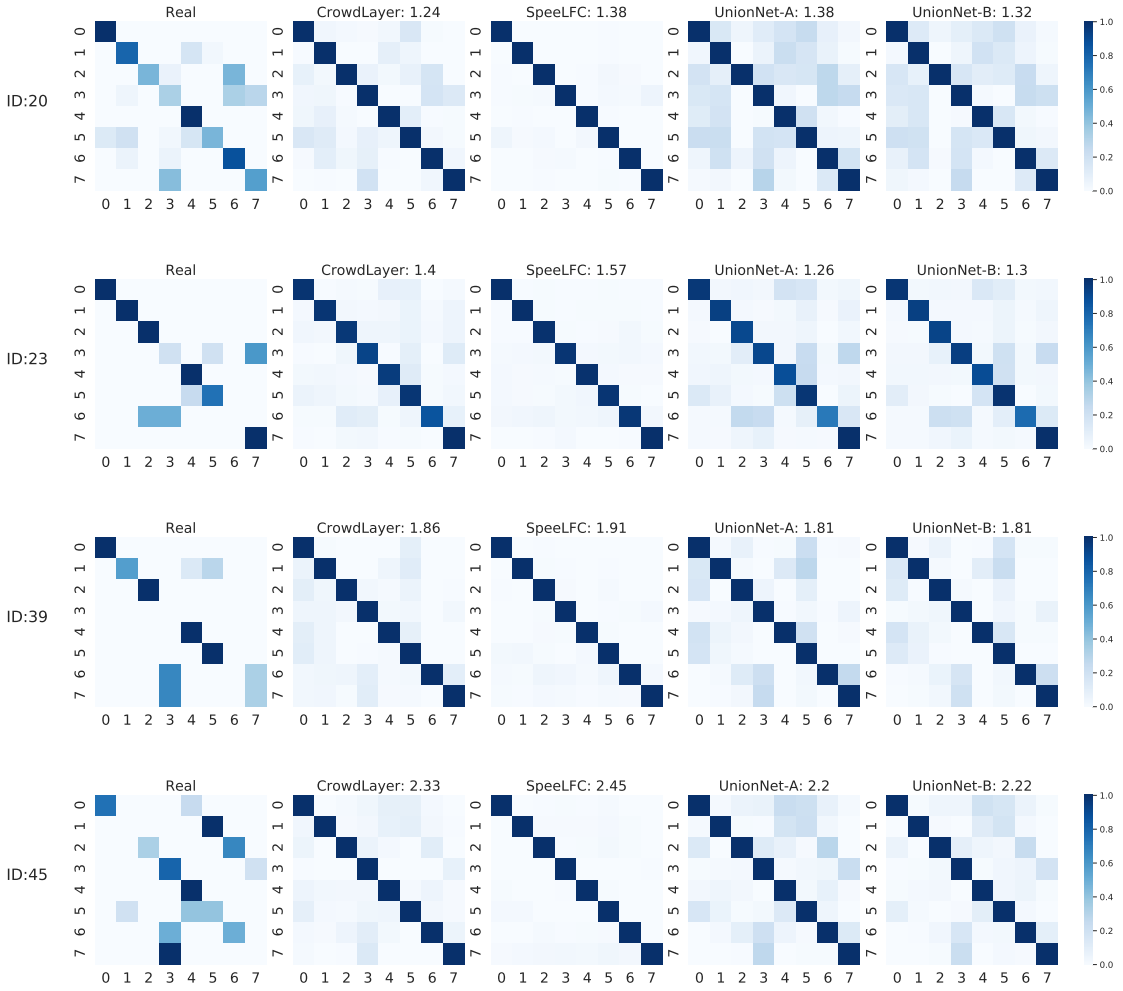
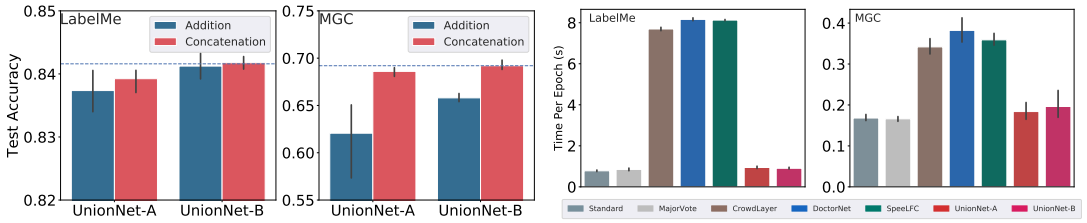


FIGURE 4.2: Four examples in LabelMe for comparisons between the learned transition matrix of different algorithms and the real transition matrix. Note that the color intensity of the cells increases with the relative magnitude. The value behind the name of each algorithm is defined as $\|\mathbf{T}^j_{\text{alg}} - \mathbf{T}^j_{\text{real}}\|_{\text{F}}$, where $\mathbf{T}^j_{\text{alg}}$ and $\mathbf{T}^j_{\text{real}}$ denote the learned weight matrix and the corresponding real confusion matrices for the jth annotator. The smaller the value, the better the fitting performance.

4.5.5 Ablation study

To demonstrate the advantage of our concatenation operation over the addition operation, we set up the experiments above LabelMe and MGC datasets. For reimplementing UnionNet-A and UnionNet-B with an addition operation, we initialize the square transition matrix by Eq. (4.7) on LabelMe. Slightly different from Eq. (8), the transition matrix on MGC is initialized by taking logarithm before addition. The results are presented in Figure 4.3a. On both LabelMe and



(A) Variants (B) Running time

FIGURE 4.3: (A) Average test accuracy (%) with standard deviation on LabelMe (left) and MGC (right) over 5 trials, for different implementations of UnionNet. (B) Average training time per epoch (over 10 epochs) on LabelMe (left) and MGC (right) for different algorithms.

MGC datasets, UnionNet-A and UnionNet-B with concatenation operation consistently perform better than the corresponding algorithms based on the addition operation, respectively. It verifies that, compared to the addition operation, the concatenation operation is more effective to build the union of crowdsourced labels for training.

4.5.6 Training efficiency analysis

To demonstrate the advantage of UnionNet in training efficiency, we train each algorithm for 10 epochs on real-world datasets and show the average training time per epoch in Figure 4.3b. Besides, we introduce a new baseline named Standard, which simply trains on the ground true label. We omit WDN in the comparisons as it is one of the variants of DoctorNet and its training time is much higher than that of the DoctorNet.

On LabelMe dataset, CrowdLayer, DoctorNet and SpeeLFC take 8-9 times longer than Standard and MajorVote. In contrast, the training speed of both UnionNet-A and UnionNet-B is close to that of Standard and MajorVote, which means the linear layer proposed in our algorithms runs with almost no additional computational burden.

The same phenomenon happens on MGC dataset. Due to their for-loop forward propagation in multiple annotator-specific transition matrices, DoctorNet spends the most time in training, followed by CrowdLayer and SpeeLFC. On the other hand, learning directly from the concatenated labels through a simple linear layer

costs very little on computational resources, shown by UnionNet-A and UnionNet-B. It shows that our proposed UnionNet is an end-to-end learning architecture with higher computational efficiency compared with other existing end-to-end learning algorithms.

4.6 Chapter Summary

In this chapter, we investigated the problem of learning with crowdsourced labels and proposed a novel end-to-end deep learning method named UnionNet, which is not only *theoretically consistent* but also *experimentally effective and efficient*. Specifically, unlike existing methods that either fit the given label by each annotator independently or fuse all the labels into a reliable one, we concatenated the one-hot encoded vectors of crowdsourced labels provided by all the annotators, which takes all the labeling information as a union and keeps it intact and coordinates multiple annotators. In this way, we could directly train an end-to-end deep neural network by maximizing the likelihood of this union with only a parametric transition matrix. We theoretically proved the learning consistency and experimentally showed the effectiveness and the efficiency of our proposed method. In future work, we will extend UnionNet to the regression and structured prediction problems.

Chapter 5

Improving OOD Detection with Logit Normalization

5.1 Motivation

Modern neural networks deployed in the open world often struggle with out-of-distribution (OOD) inputs—samples from a different distribution that the network has not been exposed to during training, and therefore should not be predicted with high confidence at test time. A reliable classifier should not only accurately classify known in-distribution (ID) samples, but also identify as "unknown" any OOD input. This gives rise to the importance of OOD detection, which determines whether an input is ID or OOD and allows the model to take precautions in deployment.

A naive solution uses the maximum softmax probability (MSP)—also known as the softmax confidence score—for OOD detection [115]. The operating hypothesis is that OOD data should trigger relatively lower softmax confidence than that of ID data. Whilst intuitive, the reality shows a non-trivial dilemma. In particular, deep neural networks can easily produce overconfident predictions, *i.e.*, abnormally high softmax confidences, even when the inputs are far away from the training data [191]. This has cast significant doubt on using softmax confidence for OOD detection. Indeed, many prior works turned to define alternative OOD scoring functions [31, 116, 117, 120, 123, 124, 192]. Yet to date, the community still has a limited understanding of the fundamental cause and mitigation of the overconfidence issue.

In this work, we show that the overconfidence issue can be mitigated through a simple fix to the cross-entropy loss—the most commonly used training objective for classification—by enforcing a constant norm on the logit vector (*i.e.*, pre-softmax output). Our method, *Logit Normalization* (dubbed **LogitNorm**), is motivated by our analysis on the norm of the neural network's logit vectors. We find that even when most training examples are classified to their correct labels, the softmax cross-entropy loss can continue to increase the magnitude of the logit vectors. The growing magnitude during training thus leads to the overconfidence issue, despite having no improvement on the classification accuracy.

To mitigate the issue, our key idea behind LogitNorm is to decouple the influence of output's norm from the training objective and its optimization. This can be achieved by normalizing the logit vector to have a constant norm during training. In effect, our LogitNorm loss encourages the direction of the logit output to be consistent with the corresponding one-hot label, without exacerbating the magnitude of the output. Trained with normalized outputs, the network tends to give conservative predictions and results in strong separability of softmax confidence scores between ID and OOD inputs (see Figure 5.4).

Extensive experiments demonstrate the superiority of LogitNorm over existing methods for OOD detection. First, our method drastically improves the OOD detection performance using the softmax confidence score. For example, using CIFAR-10 dataset as ID and SVHN as OOD data, our approach reduces the FPR95 from 50.33% to 8.03%—a **42.30**% of improvement over the baseline [115]. Averaged over a diverse collection of OOD datasets, our method reduces the FPR95 by **33.87**% compared to using the softmax score with cross-entropy loss. Beyond MSP, we show that our method not only outperforms, but also boosts more advanced post-hoc OOD scoring functions, such as ODIN [116], energy score [31], and GradNorm score [117]. In addition to the OOD detection task, our method improves the calibration performance on the ID data itself by way of post-hoc temperature scaling.

Overall, *using LogitNorm loss achieves strong performance on OOD detection and calibration tasks while maintaining the classification accuracy on ID data.* Our method can be easily adopted in practice. It is straightforward to implement with existing deep learning frameworks, and does not require sophisticated changes to the loss or training scheme.

We summarize our contributions as follows:

1. We introduce LogitNorm [1] – a simple and effective alternative to the cross-entropy loss, which decouples the influence of logits' norm from the training procedure. We show that LogitNorm can effectively generalize to different network architectures and boost different post-hoc OOD detection methods.
2. We conduct extensive evaluations to show that LogitNorm can improve both OOD detection and confidence calibration while maintaining the classification accuracy on ID data. Compared with the cross-entropy loss, LogitNorm achieves an FPR95 reduction of 33.87% on common benchmarks with the softmax confidence score.
3. We perform ablation studies that lead to improved understandings of our method. In particular, we contrast with alternative methods (*e.g.*, GODIN [119], Logit Penalty) and demonstrate the advantages of LogitNorm. We hope that our insights inspire future research to further explore loss function design for OOD detection.

5.2 Preliminaries

5.2.1 Out-of-distribution detection

Setup. We consider a supervised multi-class classification problem. We denote by $\mathcal{X}$ the input space and $\mathcal{Y} = \{1, \dots, k\}$ the label space with k classes. The training dataset $\mathcal{D}_{\text{train}} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{N}$ consists of N data points, sampled *i.i.d.* from a joint data distribution $\mathcal{P}_{\mathcal{X}\mathcal{Y}}$. We use $\mathcal{P}_{\text{in}}$ to denote the marginal distribution over $\mathcal{X}$, which represents the in-distribution (ID). Given the training dataset, we learn a classifier $f : \mathcal{X} \to \mathbb{R}^k$ with trainable parameter $\theta \in \mathbb{R}^p$, which maps an input to the output space. An ideal classifier can be obtained by minimizing the following expected risk:

$$\mathcal{R}_{\mathcal{L}}(f) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{P}_{\mathcal{X}\mathcal{Y}}}\left[\mathcal{L}\left(f(\boldsymbol{x};\theta), y\right)\right],$$

[1] The work in this chapter has been published in [193].

where $\mathcal{L}$ is the commonly used cross-entropy loss with the softmax activation function:

$$\mathcal{L}_{\text{CE}}(f(\boldsymbol{x};\theta), y) = -\log p(y|\boldsymbol{x}) = -\log \frac{e^{f_y(\boldsymbol{x};\theta)}}{\sum_{i=1}^{k} e^{f_i(\boldsymbol{x};\theta)}}.$$

Here, $f_y(\boldsymbol{x};\theta)$ denotes the y-th element of $f(\boldsymbol{x};\theta)$ corresponding to the ground-truth label y, and $p(y|\boldsymbol{x})$ is the corresponding softmax probability.

Problem statement. During the deployment time, it is ideal that the test data are drawn from the same distribution $\mathcal{P}_{\text{in}}$ as the training data. However, in reality, inputs from unknown distributions can arise, whose label set may have no intersection with $\mathcal{Y}$. Such inputs are termed out-of-distribution (OOD) data and should not be predicted by the model.

The OOD detection task can be formulated as a binary classification problem: determining whether an input $\boldsymbol{x} \in \mathcal{X}$ is from $\mathcal{P}_{\text{in}}$ (ID) or not (OOD). OOD detection can be performed by a level-set estimation:

$$g(\boldsymbol{x}) = \begin{cases} \text{in} & \text{if } S(\boldsymbol{x}) \geq \gamma \\ \text{out} & \text{if } S(\boldsymbol{x}) < \gamma \end{cases}, \tag{5.1}$$

where $S(\boldsymbol{x})$ denotes a scoring function and γ is commonly chosen so that a high fraction (e.g., 95%) of ID data is correctly classified. By convention, samples with higher scores $S(\boldsymbol{x})$ are classified as ID and vice versa. In Section 5.4.2, we will consider a variety of popular OOD scoring functions including MSP [115], ODIN [116], energy score [31] and GradNorm [117].

5.3 The Proposed Approach

5.3.1 Why models can be overconfident?

In the following, we investigate why neural networks trained with the common softmax cross-entropy loss tend to give overconfident predictions. Our analysis suggests that the large magnitude of neural network output can be a culprit.

For notation shorthand, we use $\boldsymbol{f}$ to denote the network output $f(\boldsymbol{x};\theta)$ for an input $\boldsymbol{x}$. $f(\boldsymbol{x};\theta)$ is also known as the logit, or pre-softmax output. Without loss of generality, the logit vector $\boldsymbol{f}$ can be decomposed into two components:

$$\boldsymbol{f} = \|\boldsymbol{f}\| \cdot \hat{\boldsymbol{f}}, \tag{5.2}$$

where $\|\boldsymbol{f}\| = \sqrt{\boldsymbol{f}_1^2 + \boldsymbol{f}_2^2 + \cdots + \boldsymbol{f}_k^2}$ is the Euclidean norm of the logit vector $\|\boldsymbol{f}\|$, and $\hat{\boldsymbol{f}}$ is the unit vector in the same direction as $\boldsymbol{f}$. In other word, $\|\boldsymbol{f}\|$ and $\hat{\boldsymbol{f}}$ represent the *magnitude* and the *direction* of the logit vector $\boldsymbol{f}$, respectively.

During the test stage, the model makes class predictions by $c = \arg\max_i(f_i)$. We have the following propositions.

Proposition 5.1. *For any given constant value $s > 1$, if $\arg\max_i(f_i) = c$, then $\arg\max_i(sf_i) = c$ always holds.*

Given the above proposition, we find that scaling the magnitude $\|\boldsymbol{f}\|$ of the logit does not change the predicted class c. In the following, we further explore how it impacts the softmax confidence score.

Proposition 5.2. *For the softmax cross-entropy loss, let σ be the softmax activation function. For any given scalar $s > 1$, if $c = \arg\max_i(f_i)$, then $\sigma_c(s\boldsymbol{f}) \geq \sigma_c(\boldsymbol{f})$ holds.*

The proofs of the above propositions are presented in Appendix B.1 and B.2. From Proposition 5.2, we find that increasing the magnitude $\|\boldsymbol{f}\|$ will cause a higher value for the softmax confidence score but leave the final prediction unchanged.

To analyze the impact on training objective, we provide the following formulation according to Eq. (5.2):

$$\mathcal{L}_{\text{CE}}(f(\boldsymbol{x};\theta), y) = -\log p(y|\boldsymbol{x}) = -\log \frac{e^{\|\boldsymbol{f}\| \cdot \hat{f}_y}}{\sum_{i=1}^{k} e^{\|\boldsymbol{f}\| \cdot \hat{f}_i}}.$$

We can find that the training loss depends on the magnitude $\|\boldsymbol{f}\|$ and the direction $\hat{\boldsymbol{f}}$. By keeping the direction unchanged, we analyze the influence of the magnitude $\|\boldsymbol{f}\|$ on the training loss. When $y = \arg\max_i(f_i)$, we can see that increasing $\|\boldsymbol{f}\|$ would increase $p(y|\boldsymbol{x})$. It implies that, for those training examples that are already classified correctly, the optimization on the training loss would further increase

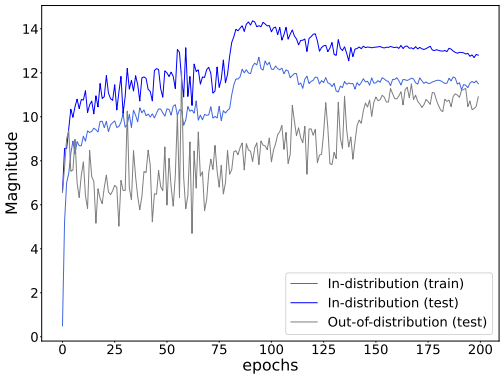


FIGURE 5.1: The mean magnitudes of logits under different training epochs. Model is trained on CIFAR-10 with WRN-40-2 [1]. OOD examples are from SVHN dataset.

the magnitude $\|\boldsymbol{f}\|$ of the network output to produce a higher softmax confidence score, thus obtaining a smaller loss.

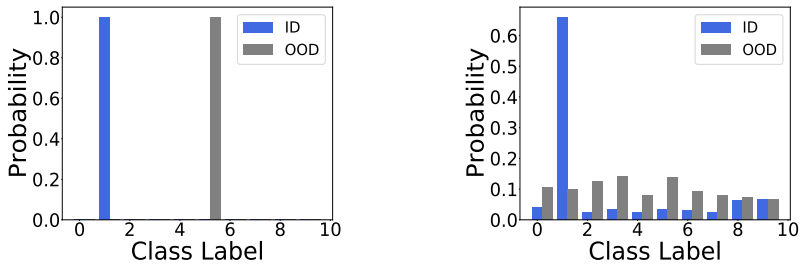
To provide a straightforward view, we show in Figure 5.1 the dynamics of logit norm during training. Indeed, softmax cross-entropy loss encourages the model to produce logits with increasingly larger norms for both ID and OOD examples. The large norms directly translate into overconfident softmax scores, leading to difficulty in separating ID vs. OOD data. We proceed by introducing our method, targeting this problem.

5.3.2 Method

In our previous analysis, we show that the softmax cross-entropy loss encourages the network to produce logits with larger magnitudes, leading to the overconfidence issue that makes it difficult to distinguish ID and OOD examples. To alleviate this issue, our key idea is to decouple the influence of logits' magnitude from network optimization. In other words, our goal is to keep the L_2 vector norm of logits a constant during the training. Formally, the objective can be formulated as:

$$\begin{aligned} \text{minimize} \quad & \mathbb{E}_{\mathcal{P}_{\mathcal{X}\mathcal{Y}}} \left[\mathcal{L}_{\text{CE}} \left(f(\boldsymbol{x}; \theta), y \right) \right] \\ \text{subject to} \quad & \|f(\boldsymbol{x}; \theta)\|_2 = \alpha. \end{aligned}$$

Performing constrained optimization in the context of modern neural networks is non-trivial. As we will show later in Section 5.5, simply adding the constraint via



(A) Cross-entropy Loss (B) LogitNorm Loss (ours)

FIGURE 5.2: The softmax outputs of two examples on a CIFAR-10 pre-trained WRN-40-2 [1] with (a) cross-entropy loss and (b) logit normalization loss. For the cross-entropy loss, the softmax confidence scores are 1.0 and 1.0 for the ID and OOD examples. In contrast, the softmax confidence scores of the network trained with LogitNorm loss are 0.66 and 0.14 for the ID and OOD examples. While using cross-entropy loss produces an extremely confident prediction for the OOD example, our method produces almost uniform softmax probabilities (near 0.1), which benefits OOD detection.

Lagrange multiplier [194] does not work well. To circumvent the issue, we convert the objective into an alternative loss function that can be end-to-end trainable, which strictly enforces a constant vector norm.

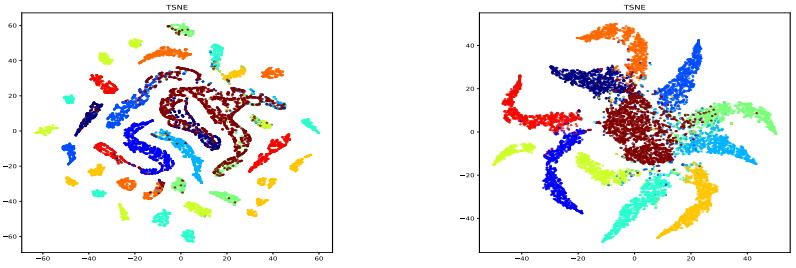
Logit Normalization. We employ *Logit Normalization* (dubbed LogitNorm), which encourages the direction of the logit to be consistent with the corresponding one-hot label, without optimizing the magnitude of the logit. In particular, the logit vector is normalized to be a unit vector with a constant magnitude. The softmax cross-entropy loss is then applied on the normalized logit vector instead of the original output. Formally, the objective function of LogitNorm is given by:

$$\mathcal{R}_{\mathcal{L}}(f) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{P}_{\mathcal{X}\mathcal{Y}}}\left[\mathcal{L}_{\text{CE}}\left(\hat{f}(\boldsymbol{x};\theta), y\right)\right], \tag{5.3}$$

where $\hat{f}(\boldsymbol{x};\theta) = f(\boldsymbol{x};\theta)/\|f(\boldsymbol{x};\theta)\|$ is the normalized logit vector. In practice, a small positive value (e.g., 10^{-7}) is added to the denominator to ensure numerical stability. Equivalently, the new loss function can be defined as:

$$\mathcal{L}_{\text{logit\_norm}}(f(\boldsymbol{x};\theta), y) = -\log \frac{e^{f_y/(\tau\|\boldsymbol{f}\|)}}{\sum_{i=1}^{k} e^{f_i/(\tau\|\boldsymbol{f}\|)}}, \tag{5.4}$$

where the temperature parameter τ modulates the magnitude of the logits. Interestingly, our loss function can be viewed as having an *input-dependent* temperature, $\tau\|f(\boldsymbol{x};\theta)\|$, which depends on the input $\boldsymbol{x}$.



(A) Cross-entropy Loss (B) LogitNorm Loss (ours)

FIGURE 5.3: t-SNE visualization [2] of the softmax outputs from WRN-40-2 [1] trained on CIFAR-10 with (a) cross-entropy loss and (b) LogitNorm loss. All colors except for brown indicate 10 different ID classes. Points in *brown* denote out-of-distribution examples from SVHN. Trained with logit normalization, the softmax outputs provide more meaningful information to differentiate in- and out-distribution samples.

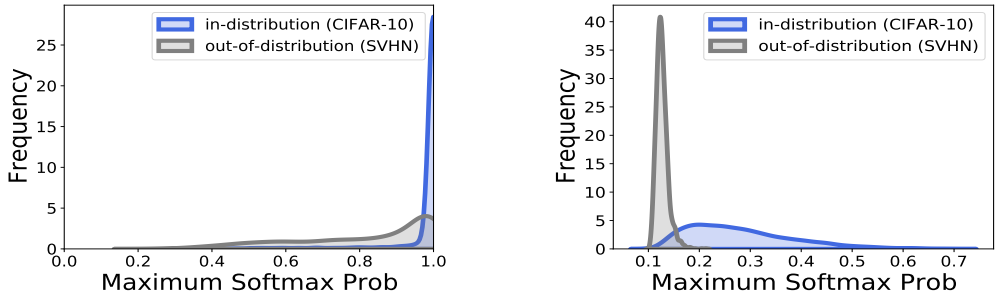
By way of logit normalization, the magnitudes of the output vectors are strictly constant (*i.e.*, $1/\tau$). Minimizing the loss in Eq. (5.4) can only be achieved by adjusting the direction of the logit output f. The resulting model tends to give conservative predictions, especially for inputs that are far away from $\mathcal{P}_{\text{in}}$. We illustrate with an example in Figure 5.2, where training with logit normalization leads to softmax outputs that are more distinguishable between in- and out-of-distribution samples (right), as opposed to using cross-entropy loss (left). In Figure 5.3, we present a t-SNE visualization of the softmax outputs using Cross-entropy vs. LogitNorm Loss, where LogitNorm leads to more meaningful information to differentiate ID and OOD samples in the softmax output space. Below we further provide a lower bound of this new loss function in Eq. (5.4).

Proposition 5.3 (Lower Bound of Loss)**.** *For any input $\boldsymbol{x}$ and any positive number $\tau \in \mathbb{R}^+$, the per-sample loss defined in Eq. (5.4) has a lower bound: $\mathcal{L}_{logit\_norm} \geq \log\left(1 + (k-1)e^{-2/\tau}\right)$, where k is the number of classes.*

The proof of Proposition 5.3 is provided in Appendix B.3. From Proposition 5.3, we find that the LogitNorm loss has a lower bound that depends on τ and number of classes k. In particular, it implies that the lower bound of the loss value increases with the value of τ. For example, when $k = 10$ and $\tau = 1$, the norm of logits would be linearly scaled to 1 and the lower bound is about 0.7966. The high lower bound can cause optimization difficulty. For this reason, we found it is desirable to have a relatively small $\tau < 1$. We will analyze the effect of τ in detail in Section 5.5.

TABLE 5.1: OOD detection performance comparison using softmax cross-entropy loss and LogitNorm loss. We use WRN-40-2 [1] to train on the in-distribution datasets and use softmax confidence score as the scoring function. All values are percentages. ↑ indicates larger values are better, and ↓ indicates smaller values are better. **Bold** numbers are superior results.

ID datasets	CIFAR-10			CIFAR-100		
OOD datasets	FPR95 ↓	AUROC ↑	AUPR ↑	FPR95 ↓	AUROC ↑	AUPR ↑
	Cross-entropy loss / **LogitNorm loss (ours)**					
Texture	64.13 / **28.64**	86.29 / **94.28**	96.28 / **98.63**	84.11 / **70.67**	74.05 / **78.65**	93.45 / **93.66**
SVHN	50.33 / **8.03**	93.48 / **98.47**	98.70 / **99.68**	79.09 / **45.98**	78.62 / **92.48**	94.99 / **98.45**
LSUN-C	33.34 / **2.37**	95.29 / **99.42**	99.04 / **99.88**	67.94 / **13.93**	83.60 / **97.56**	96.32 / **99.48**
LSUN-R	42.52 / **10.93**	93.74 / **97.87**	98.66 / **99.59**	82.21 / **68.68**	69.45 / **84.77**	91.45 / **96.52**
iSUN	46.56 / **12.28**	93.13 / **97.73**	98.55 / **99.56**	84.50 / **71.47**	69.29 / **83.79**	91.44 / **96.24**
Places365	60.23 / **31.64**	87.36 / **93.66**	96.62 / **98.49**	81.09 / **80.20**	75.61 / **77.14**	93.74 / **94.16**
average	49.52 / **15.65**	91.55 / **96.91**	97.98 / **99.31**	79.82 / **58.49**	75.10 / **85.73**	93.57 / **96.42**



(A) Cross-entropy Loss (B) LogitNorm Loss (Ours)

FIGURE 5.4: Distribution of softmax confidence score from WRN-40-2 [1] trained on CIFAR-10 with (a) cross-entropy loss and (b) LogitNorm loss.

5.4 Experiments

In this section, we verify the effectiveness of LogitNorm loss in OOD detection with several benchmark datasets.

5.4.1 Experimental setup

In-distribution datasets. In this work, we use the CIFAR-10 and CIFAR-100 [84] datasets as in-distribution datasets, which are common benchmarks for OOD detection. Specifically, we use the standard split with 50,000 training images and 10,000 test images. All the images are of size 32×32.

TABLE 5.2: OOD detection performance comparison with various scoring functions using softmax cross-entropy loss and logit normalization loss. We use WRN-40-2 to train on the in-distribution datasets. All values are percentages. ↑ indicates larger values are better, and ↓ indicates smaller values are better. **Bold** numbers are superior results.

ID datasets	CIFAR-10			CIFAR-100		
Score	FPR95 ↓	AUROC ↑	AUPR ↑	FPR95 ↓	AUROC ↑	AUPR ↑
	Cross-entropy loss / **LogitNorm loss (ours)**					
Softmax	49.52 / **15.65**	91.55 / **96.91**	97.98 / **99.31**	79.82 / **58.35**	75.10 / **85.75**	93.57 / **96.44**
ODIN	40.32 / **12.95**	87.21 / **97.37**	96.24 / **99.40**	70.71 / **58.13**	81.38 / **85.65**	95.37 / **96.36**
Energy	26.82 / **19.14**	93.07 / **96.09**	98.02 / **99.14**	70.87 / **65.46**	81.45 / **84.84**	95.38 / **96.27**
GradNorm	58.98 / **17.78**	72.29 / **96.34**	90.75 / **99.14**	87.01 / **61.89**	52.84 / **81.41**	84.12 / **94.85**

Out-of-distribution datasets. For the OOD detection evaluation, we use six common benchmarks as OOD test datasets $\mathcal{D}_{\text{out}}^{\text{test}}$: `Textures` [195], `SVHN` [196], `Places365` [86], `LSUN-Crop` [197], `LSUN-Resize` [197], and `iSUN` [198]. For all test datasets, the images are of size 32×32.

Evaluation metrics. We evaluate the performance of OOD detection by measuring the following metrics: (1) the false positive rate (FPR95) of OOD examples when the true positive rate of in-distribution examples is 95%; (2) the area under the receiver operating characteristic curve (AUROC); and (3) the area under the precision-call curve (AUPR).

Training details. For main results, we perform training with WRN-40-2 [1] on CIFAR-10/100. The network is trained for 200 epochs using SGD with a momentum of 0.9, a weight decay of 0.0005, a dropout rate of 0.3, and a batch size of 128. We set the initial learning rate as 0.1 and reduce it by a factor of 10 at 80 and 140 epochs. The hyperparameter τ is selected from the range $\{0.001, 0.005, 0.01, \ldots, 0.05\}$. We set 0.04 for CIFAR-10 by default. For hyperparameter tuning, we use Gaussian noises as the validation set. All experiments are repeated five times with different seeds, and we report the average performance. We conduct all the experiments on NVIDIA GeForce RTX 3090 and implement all methods with default parameters using PyTorch [199].

5.4.2 Results

How does logit normalization influence OOD detection performance? In Table 5.1, we compare the OOD detection performance on models trained with

TABLE 5.3: OOD detection performance comparison trained on CIFAR-10 with different network architectures: WRN-40-2 [1], ResNet-34 [4], DenseNet-BC [5]. All values are percentages. ↑ indicates larger values are better, and ↓ indicates smaller values are better. **Bold** numbers are superior results.

Architecture	ID Accuracy ↑	FPR95 ↓	AUROC ↑	AUPR ↑
	Cross-entropy loss / **LogitNorm loss (ours)**			
WRN-40-2	94.75 / 94.69	49.52 / **15.65**	91.55 / **96.91**	97.98 / **99.31**
ResNet-34	95.01 / 95.14	47.74 / **15.82**	91.15 / **97.01**	97.72 / **99.32**
DenseNet	94.55 / 94.37	50.41 / **18.57**	91.48 / **96.16**	98.04 / **99.10**

cross-entropy loss and LogitNorm loss respectively. To isolate the effect of the loss function in training, we keep the test-time OOD scoring function to be the same, *i.e.*, softmax confidence score:

$$S(\boldsymbol{x}) = \max_i \frac{e^{f_i(\boldsymbol{x};\theta)}}{\sum_{j=1}^{k} e^{f_j(\boldsymbol{x};\theta)}}.$$

A salient observation is that our method drastically improves the OOD detection performance by employing LogitNorm loss. For example, on the CIFAR-10 model, when evaluated against SVHN as OOD data, our method reduces the FPR95 from 50.33% to 8.03%—a **42.3**% of direct improvement. Averaged across six test datasets, our approach reduces the FPR95 by **33.87**% compared with using MSP on the model trained with cross-entropy loss. On CIFAR-100, our method also improves the performance by a meaningful margin.

To further illustrate the difference between the two losses on OOD detection, we visualize and compare the distribution of softmax confidence score for ID and OOD data, derived from networks trained with cross-entropy vs. LogitNorm losses. With cross-entropy loss, the softmax scores for both ID and OOD data concentrate on high values, as shown in Figure 5.4a. In contrast, the network trained with LogitNorm loss produces highly distinguishable scores between ID and OOD data. From Figure 5.4b, we observe that the softmax confidence score for most OOD examples is around 0.1, which indicates that the softmax outputs are close to a uniform distribution. Overall the experiments show that training with LogitNorm loss makes the softmax scores more distinguishable between in- and out-of-distributions and consequently enables more effective OOD detection.

Can the logit normalization improve existing scoring functions? In Table 5.2, we show that the LogitNorm loss not only outperforms, but also boosts competitive OOD scoring functions. Note that all the OOD scoring functions considered are originally developed based on models trained with cross-entropy loss, hence are natural choices of comparison. In particular, we consider: 1) **MSP** [115] uses the softmax confidence score to detect OOD samples. 2) **ODIN** [116] employs temperature scaling and input perturbation to improve OOD detection. Following the original setting, we use the temperature parameter $T = 1000$ and $\epsilon = 0.0014$. 3) **Energy score** [31] utilizes the information in logits for OOD detection, which is the negative log of the denominator in softmax function: $E(\mathbf{x}; f) = -T \cdot \log \sum_{i=1}^{k} e^{f_i(\boldsymbol{x})/T}$. With LogitNorm loss, we set $T = 0.1$ for CIFAR-10 and $T = 0.01$ for CIFAR-100. 4) **GradNorm** [117] detects OOD inputs by utilizing information extracted from the gradient space.

Our results in Table 5.2 suggest that logit normalization can benefit a wide range of downstream OOD scoring functions. Due to space constraints, we report the average performance across six test OOD datasets. For example, we observe that the FPR95 of the ODIN method is reduced from 40.32% to 12.95% when employing logit normalization, establishing strong performance. In addition, we find that logit normalization enables the energy score and GradNorm score to achieve decent OOD detection performance as well.

Logit normalization is effective on different architectures. In Table 5.3, we show that LogitNorm is effective on a diverse range of model architectures. The results are based on softmax confidence score as test-time OOD score. In particular, our method consistently improves the performance using WRN-40-2 [1], ResNet [4] and DenseNet [5] architectures. For example, on DenseNet, using LogitNorm loss reduces the average FPR95 from 50.41% to 18.57%.

Logit normalization maintains classification accuracy. We also verify whether LogitNorm loss affects the classification accuracy. Our results in Table 5.3 show that LogitNorm can improve the OOD detection performance, and at the same time, achieve similar accuracy as the cross-entropy loss. For example, when trained on ResNet-34 with CIFAR-10 as the ID dataset, using LogitNorm loss achieves a test accuracy of 95.14% on CIFAR-10, on par with the test accuracy 95.01% using cross-entropy loss. On CIFAR-100, LogitNorm loss and cross-entropy loss also achieves comparable test accuracies (75.12% vs. 75.23%). Overall LogitNorm loss

TABLE 5.4: Comparison results of ECE (%) with $M = 15$, using WRN-40-2 trained on CIFAR-10. For temperature scaling (TS), T is tuned on a hold-out validation set by optimization methods.

Dataset		Cross Entropy	LogitNorm (ours)
CIFAR-10	w/o TS	3.20	66.95
	w/ TS	0.66	**0.41**
CIFAR-100	w/o TS	11.69	70.45
	w/ TS	2.18	**1.67**

maintains comparable classification accuracy on the ID data while leading to significant improvement in OOD detection performance.

Logit normalization enables better calibration. While the OOD detection task concentrates on the separability between ID and OOD data, the calibration task focuses solely on the ID data—softmax confidence score should represent the true probability of correctness [200]. In practice, Expected Calibration Error (ECE) [201] is commonly used to measure the calibration performance from finite samples. In Figure 5.4, we observe that LogitNorm loss leads to a smoother distribution of softmax confidence score for ID examples, in contrast to the spiky distribution induced by cross-entropy loss (*i.e.*, values concentrate around 1). It implies that the model trained with LogitNorm loss preserves distinguishable information for different ID samples, indicating its potential in improving model calibration. Indeed, our results in Table 5.4 show that the model trained with LogitNorm achieves better calibration performance by way of post-hoc temperature scaling.

5.5 Discussion

Logit normalization vs. Logit penalty. While our logit normalization has demonstrated strong promise, a question arises: *can a similar effect be achieved by imposing a penalty on the L_2 norm of the logits*? In this ablation, we show that explicitly constraining logit norm via a Lagrangian multiplier [194] does not work well. Specifically, we consider the alternative loss:

$$\mathcal{L}_{\text{logit\_penalty}}(f(\boldsymbol{x};\theta), y) = \mathcal{L}_{\text{CE}}(f(\boldsymbol{x};\theta), y) + \lambda \|f(\boldsymbol{x};\theta)\|_2.$$

TABLE 5.5: OOD detection performance comparison with different loss functions. We use WRN-40-2 trained on CIFAR-10. **Bold** numbers are superior results. We report the average norm value of logits as L_2 Norm, where we use SVHN as OOD test dataset.

Loss	FPR95 ↓	AUROC ↑	AUPR ↑	L_2 Norm (ID / OOD)
CrossEntropy	49.52	91.55	97.98	12.80 / 10.91
LogitPenalty	57.62	73.27	87.62	1.90 / 1.74
LogitNorm (ours)	**15.65**	**96.91**	**99.31**	1.48 / 0.49

TABLE 5.6: Comparison between GODIN and LogitNorm. We use WRN-40-2 [1] trained on CIFAR-10. For a fair comparison, we use the ODIN score for LogitNorm loss. **Bold** numbers are superior results.

Loss	FPR95 ↓	AUROC ↑	AUPR ↑
GODIN	25.24	95.07	98.93
LogitNorm (ours)	**15.65**	**96.91**	**99.31**

where λ denotes the Lagrangian multiplier that controls the trade-off between the cross-entropy loss and the regularization term.

Our results in Table 5.5 show that both logit penalty and logit normalization lead to logits with small L_2 norms, compared with using cross-entropy loss. However, unlike LogitNorm, the logit penalty method produces a large L_2 norm for OOD data as well, resulting in the inferior performance of OOD detection. In practice, we notice that the network trained with the logit penalty can suffer from optimization difficulty and sometimes fail to converge if λ is too large (which is needed to regularize the logit norm effectively). Overall, we show that simply constraining the logit norm during training cannot help the OOD detection task while our LogitNorm loss significantly improves the performance.

Relations to temperature scaling. As introduced in Section 5.3, the logit normalization can be viewed as an *input-dependent* temperature on the logits. Related to our work, previous work ODIN [116] proposes a variant of softmax score by employing temperature scaling with a *constant* T in the testing phase:

$$S(\boldsymbol{x}) = \max_i \frac{e^{f_i(\boldsymbol{x};\theta)/T}}{\sum_{j=1}^{k} e^{f_j(\boldsymbol{x};\theta)/T}},$$

where the temperature is the same for all inputs. In contrast, our method bares two key differences: (1) the effective temperature in LogitNorm is input-dependent

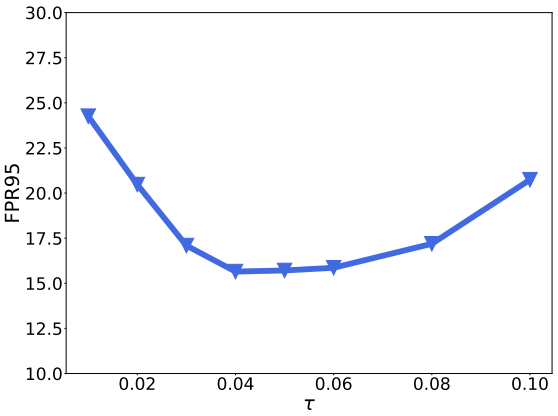


FIGURE 5.5: Effect of τ in LogitNorm with CIFAR-10.

rather than a global constant, and (2) the temperature in LogitNorm can be enforced during the training stage. Our method is compatible with test-time temperature scaling in OOD detection and can improve calibration performance with temperature scaling, as we show in Section 5.4.2.

In Figure 5.5, we further ablate how the parameter τ in our method (*cf.* Eq. 5.4) affects the OOD detection performance. The analysis is based on CIFAR-10, with FPR95 averaged across six test datasets. Our results echo the analysis in Proposition 5.3, where a large τ would lead to a large lower bound on the loss, which is less desirable from the optimization perspective.

Relations to other normalization methods. In the literature, normalization method is applied for OOD detection in the form of cosine similarity [119, 202]. In particular, Cosine loss [202] and Generalized ODIN (GODIN) [119] decompose the logit $f_i(\mathbf{x};\theta)$ for class i as: $f_i(\mathbf{x};\theta) = \frac{h_i(\mathbf{x};\theta)}{g(\mathbf{x};\theta)}$, where

$$\begin{cases} h_i(\boldsymbol{x}) = \cos(\boldsymbol{w}_i, \phi^p(\boldsymbol{x})) = \frac{\boldsymbol{w}_i^T \phi^p(\boldsymbol{x})}{\|\boldsymbol{w}_i\| \|\phi^p(\boldsymbol{x})\|}, \\ g(\mathbf{x};\theta) = \exp\left\{\text{BN}\left(\boldsymbol{W}^\top \phi^p(\boldsymbol{x}) + b\right)\right\}. \end{cases}$$

In the testing phase, the maximum cosine similarity is used as the scoring function. Contrastingly, LogitNorm has two key differences: (1) the cosine similarity in $h_i(\boldsymbol{x})$ applies L_2 normalization on the last-layer weight $\boldsymbol{w}$ and the learned feature $\phi^p(\boldsymbol{x})$, while our LogitNorm normalizes the network output $f(\boldsymbol{x};\theta)$; (2) our LogitNorm can boost the performance of common OOD scoring functions, while GODIN and Cosine loss detect OOD examples with their specific scoring functions. In Table 5.6,

we show our LogitNorm with the MSP score achieves better performance than GODIN in OOD detection.

5.6 Related Work

OOD detection. OOD detection is an increasingly important topic for deploying machine learning models in the open world and has attracted a surge of interest in two directions.

1) Some methods aim to design scoring functions for OOD detection, such as OpenMax score [118], maximum softmax probability [115], ODIN score [116, 119], Mahalanobis distance-based score [120], Energy-based score [31, 121, 122], ReAct [123], GradNorm score [117], and non-parametric KNN-based score [124, 125]. In this work, we first show that logit normalization can drastically mitigate the overconfidence issue for OOD data, thereby boosting the performance of existing scoring functions in OOD detection.

2) Some works address the out-of-distribution detection problem by training-time regularization [3, 31, 126–135]. For example, models are encouraged to give predictions with uniform distribution [3, 126] or higher energies [31, 135–138] for outliers. The energy-based regularization has a direct theoretical interpretation as shaping the log-likelihood, hence naturally suits OOD detection. Contrastive learning methods are also employed for the OOD detection task [139–141], which can be computationally more expensive to train than ours. In this work, we focus on exploring classification-based loss functions for OOD detection, which only requires in-distribution data in training. LogitNorm is easy to implement and use, and maintains the same training scheme as standard cross-entropy loss.

Normalization in deep learning. In the literature, normalization has been widely used in metric learning [203–205], face recognition [206–211], and self-supervised learning [212]. L_2-constrained softmax [206] applies the L_2 normalization on features and SphereFace [207] normalizes the weights of the last inner-product later only. Cosine loss [208, 209] normalizes both the features and weights to achieve better performance for face verification. LayerNorm [213] normalizes the distributions of intermediate layers for better generalization accuracy. In self-supervised learning, SimCLR [212] adopts cosine similarity to measure the feature

distances between positive pair of examples. A recent study [214] shows that several loss functions, including logit normalization and cosine softmax, lead to higher accuracy on ImageNet but degrade the performance of transfer tasks. In addition, GODIN [119] and Cosine loss [202] adopt cosine similarity for better performance on OOD detection. As discussed in Section 5.5, our method is superior to these cosine-based methods, since it is applicable to existing scoring functions and achieves strong performance in OOD detection.

Confidence calibration. Confidence calibration has been studied in various contexts in recent years. Some works address the miscalibration problem by post-hoc methods, such as Temperature Scaling [200, 215] and Histogram Binning [216]. Besides, some regularization methods are also proposed to improve the calibration quality of deep neural networks, like weight decay [200], label smoothing [80, 217], and focal loss [218, 219]. Conformal prediction based method [220] outputs the empty set as prediction in case of too high "atypicality". Top-label calibration aims to calibrate the reported probability for the predicted class label [221]. Recent work [222] shows that these regularization methods make it harder to further improve the calibration performance with post-hoc methods. LogitNorm loss yields better calibration performance with Temperature Scaling than cross-entropy.

5.7 Chapter Summary

In this chapter, we introduce Logit Normalization (LogitNorm), a simple alternative to the cross-entropy loss that enhances many existing post-hoc methods for OOD detection. By decoupling the influence of logits' norm from the training objective and its optimization, the model tends to give conservative predictions for OOD inputs, resulting in a stronger separability from ID data. Extensive experiments show that LogitNorm can improve both OOD detection and confidence calibration while maintaining the classification accuracy on ID data. This method can be easily adopted in practical settings. It is straightforward to implement with existing deep learning frameworks, and does not require sophisticated changes to the loss or training scheme. We hope that our insights inspire future research to further explore loss function design for OOD detection.

Chapter 6

Improving Robustness against Noisy Labels with OOD Instances

6.1 Motivation

The success of deep neural networks (DNNs) heavily relies on a large number of training instances with fully accurate labels [4]. In real-world applications, it is expensive and time-consuming to collect such large-scale datasets. To alleviate this problem, some surrogate methods are commonly used to improve labelling efficiency, such as online queries [24] and crowdsourcing [23]. These cheap but imperfect methods usually suffer from unavoidable noisy labels that can be easily memorized by Deep Neural Networks (DNNs), leading to poor generalization performance [50, 223]. Therefore, designing robust algorithms against noisy labels is of great practical importance.

Label noise can be separated into two categories: 1) closed-set noise, where instances with noisy labels have true class labels within the noisy label set [27, 59, 63, 67, 69, 160]. 2) open-set noise, where instances with noisy labels have some true class labels outside the noisy label set [32–35]. In the existing literature, open-set noises are always considered to be harmful to the training of DNNs like closed-set noises [32–35]. For example, ILON [33] reduces the effect of open-set noises by using noisy label detector and contrastive loss to pull away noisy samples from clean samples. Extended T [35] uses an extended transition matrix to mitigate the

impacts of open-set noises and EvidentialMix [34] filters out the open-set examples by modelling the loss distribution. Therefore, there arises a question to be answered: Do open-set noises always play a negative role in the training?

In this work, we answer this question by experiments with an open-set auxiliary dataset. We empirically show that additional open-set noises can be harmless to generalization if the number of open-set noises is sufficiently large. More surprisingly, we observe that the open-set noises can even benefit the robustness of neural networks against noisy labels from the original training dataset, while the closed-set counterpart could not. Based on the "insufficient capacity" hypothesis [223], an intuitive explanation for this phenomenon is that increasing the number of open-set auxiliary samples slows down the fitting of inherent noises. In other words, fitting open-set label noises consumes extra capacity of the network, thereby reducing the memorization of inherent noisy labels. The details of experimental results are presented in Subsection 6.2.2.

Inspired by the above observations, we propose an effective and complementary method to improve robustness against noisy labels. The high-level idea is to introduce Open-set samples with Dynamic Noisy Labels (ODNL [1]) into training. In each epoch, we assign random labels to instances of open-set auxiliary dataset by uniformly sampling from the label set. In other words, the label of each open-set sample is independently random and is constantly changing over training epochs. In this manner, ODNL would consistently produce "benign hard examples" to consume the extra representation capacity of neural networks, thereby preventing the neural network from overfitting inherent noisy labels.

To further understand the effect of ODNL, we provide an analysis from the lens of SGD noise [38, 224], and formalize the differences and connections between our regularization and related methods. From the analysis, we show that the SGD noises induced by our method are 1) random-direction, helping the model converge to a flat minimum; 2) conflict-free, leading to good stability for the training; 3) biased, enforcing the model to produce more conservative predictions on OOD instances. In this manner, our method not only improves the robustness against noisy labels, but also provides benefits for OOD detection performance, which is also essential for the deployment of deep learning in safety-critical applications.

[1]The work in this chapter has been published in [133].

To the best of our knowledge, we are the first to explore the benefits of open-set auxiliary dataset in learning with noisy labels. To verify the efficacy of our regularization, we conduct extensive experiments on both simulated and real-world noisy datasets, including CIFAR-10, CIFAR-100 and Clothing1M datasets. In summary, the proposed method can be easily adapted for a wide range of existing algorithms to prevent overfitting noisy labels. Furthermore, experimental results validate that our regularization could also achieve impressive performance for detecting OOD instances, even if the labels of training dataset are noisy.

6.2 Preliminaries

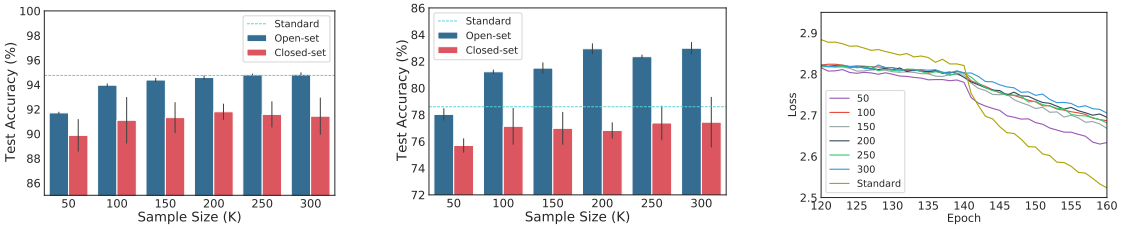
In this section, we first briefly introduce the problem setting of learning from training data with noisy labels. Then we start in Subsection 6.2.2 with an observational study highlighting the effect of open-set noisy examples from the auxiliary dataset on generalization and robustness against noisy labels.

6.2.1 learning with inherent noisy labels

In this work, we consider multi-class classification with k classes. Let $\mathcal{X} \subset \mathbb{R}^d$ be the feature space and $\mathcal{Y} = \{1, \ldots, k\}$ be the label space, we suppose the training dataset with N samples is given as $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$, where $\boldsymbol{x}_i \in \mathcal{X}$ is the i-th instance sampled from a certain data-generating distribution $\mathcal{D}_{\text{train}}$ in an i.i.d. manner and $y_i \in \{1, \ldots, k\}$ represents its observed label that might be different from the ground truth. To prevent confusion, the noisy labels in the training dataset are referred to as *inherent noisy labels*. A classifier is a function that maps the feature space to the label space $f : \mathcal{X} \to \mathbb{R}^k$ with trainable parameter $\boldsymbol{\theta} \in \mathbb{R}^p$. In this chapter, we consider the common case where the function f is a DNN with the softmax output layer. The objective function of standard training can be represented as a cross-entropy loss:

$$\mathcal{L}_{CCE} = \mathbb{E}_{\mathcal{D}_{train}} \left[\mathcal{L} \left(f(\boldsymbol{x}; \boldsymbol{\theta}), y_{\boldsymbol{x}} \right) \right] = -\frac{1}{N} \sum_{i=1}^{N} \boldsymbol{e}^{y_i} \log f \left(\boldsymbol{x}_i; \boldsymbol{\theta} \right), \tag{6.1}$$

where $f(\boldsymbol{x}_i; \boldsymbol{\theta})$ denotes the output of the classifier f with $\boldsymbol{\theta}$ for $\boldsymbol{x}_i$ and $\boldsymbol{e}^{y_i} := (0, \ldots, 1, \ldots, 0)^\top \in \{0, 1\}^k$ is a one-hot vector and only the y_i-th entry of $\boldsymbol{e}^{y_i}$ is 1.



(A) Learning with Clean data

(B) Learning with inherent noises

(C) Losses of inherent noises

FIGURE 6.1: Average test accuracy (%) on CIFAR-10 with (a) clean labels (b) inherent noisy labels under different auxiliary dataset sizes. We use "open-set" and "closed-set" to denote two types of auxiliary datasets. The results of standard training are also reported for comparison. All experiments are repeated five times and the standard deviations are presented as error bars. (c) Training losses of inherent noises around the 140th epoch under different auxiliary dataset sizes (K).

In addition to the training data, we consider there is an unlabelled auxiliary dataset consisting of open-set instances $\{\widetilde{\boldsymbol{x}}_i\}_{i=1}^{M}$. Specifically, open-set instances are sampled from $\mathcal{D}_{out}$, disjoint from $\mathcal{D}_{\text{train}}$. Therefore, they are also called Out-of-Distribution (OOD) instances. In real applications, it is easy to get such an auxiliary dataset, which is commonly used in OOD detection tasks [3, 225].

6.2.2 The effect of open-set noisy labels

To analyze the effects of open-set noisy labels, we build an open-set noisy dataset with the unlabelled auxiliary dataset. For each open-set instance, we assign a fixed random label $\widetilde{y}_i$ uniformly sampled from the label set $\{1, \ldots, k\}$. In each iteration, we concatenate the two batches that are sampled from the training dataset and the open-set noisy dataset respectively, then perform standard training. In addition, we conduct controlled experiments by replacing the open-set dataset with a closed-set dataset to verify the effectiveness of open-set noisy dataset during the training. To demonstrate the importance of the sample size of the given auxiliary dataset, we compare the performance of training with different-sized auxiliary datasets, built by randomly sampling from the original auxiliary dataset.

Throughout this subsection, we perform training with WRN-40-2 [1] on CIFAR-10 [84] using standard cross-entropy loss. For experiments under inherent label noise, we use symmetric noise with 40% noisy rate. For auxiliary dataset, we use

80 Million Tiny Images [226] as the open-set auxiliary dataset and CIFAR-5m[2] [227, 228] as the closed-set auxiliary dataset. Specifically, we remove all examples of 80 Million Tiny Images that appear in the CIFAR-10/100 datasets, so that $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{out}}$ are disjoint.

Learning with clean labels. Fig. 6.1a presents the results on clean training set of CIFAR-10, training with the open/closed-set noisy auxiliary dataset. The results show that open-set noisy labels from the auxiliary dataset become harmless gradually with the increase of the sample size of the auxiliary dataset, while closed-set noisy labels consistently deteriorate the generalization performance. Furthermore, we observe that closed-set noisy labels result in unstable training shown by the large standard deviations in the figure. It is worthy to note that open-set noisy labels do not encounter this issue.

Learning with inherent noisy labels. Fig. 6.1b presents the results on CIFAR-10 with inherent noisy labels, training with the open/closed-set noisy auxiliary dataset. Surprisingly, we can observe that as we increase the sample size of the open-set noisy dataset, the open-set noises can even improve the generalization performance beyond standard training without the noisy auxiliary dataset. In this case, closed-set noisy auxiliary dataset still plays a negative role in the training.

An intuitive interpretation. From the results, one can hypothesize that open-set noisy labels in the auxiliary dataset would be harmless and can even benefit the robustness against inherent noisy labels, if the sample size of the auxiliary dataset is large enough. This phenomenon is somewhat counter-intuitive since one would expect that open-set noisy labels should have been "poison" to the training of network, like closed-set noises [32–35]. Here we provide an intuitive interpretation from "insufficient capacity" [223]: increasing the number of examples while keeping representation capacity fixed would increase the time needed to memorize the data set. Hence, the larger the size of auxiliary dataset is, the more time it needs to memorize the open-set noises in the auxiliary dataset as well as the inherent noises in the training set, relative to clean data. This interpretation is supported by our plot of training losses of inherent noises with different-sized auxiliary datasets in Fig. 6.1c, where increasing the number of open-set auxiliary samples slows down the fitting of inherent noises.

[2]The dataset is published on `https://github.com/preetum/cifar5m`.

6.3 The Proposed Approach

Motivated by the previous observations, we propose to utilize open-set auxiliary dataset to further improve the robustness against inherent noisy labels. This approach can be considered as an auxiliary task that continuously consumes the extra capacity of neural networks but does not interfere with learning patterns from clean data. Such auxiliary tasks will prevent deep networks from overfitting noisy data.

Dynamic Noisy Labels. To maintain the effectiveness of the auxiliary task during training, an ideal method is to use an auxiliary dataset with unlimited instances based on "insufficient capacity" [223]. In this way, the network would always try to fit a stream of new instances sequentially in an online learning setting. However, it is unrealistic to provide such an auxiliary dataset in real scenarios.

To relieve the requirement on sample size, we propose an alternative approach called Dynamic Noisy Labels. For each open-set instance $\widetilde{\boldsymbol{x}}_i$, a noisy label $\widetilde{y}_i$ is uniformly sampled from the label set $\{1, \ldots, k\}$ independently, regardless of the instance $\widetilde{\boldsymbol{x}}_i$. More importantly, the generated labels are not fixed and would be changed dynamically over training epochs. In this way, the auxiliary task becomes an "impossible mission", continuously consuming the extra capacity of network. Without Dynamic Noisy Labels, the effectiveness of the auxiliary task would be degraded gradually during training, because deep network can easily memorize the open-set samples, even though the labels of those samples are randomly generated [50, 223].

Training Objective. For the original training dataset, we use the standard cross entropy as the training objective function:

$$\mathcal{L}_1 = \mathbb{E}_{\mathcal{D}_{\text{train}}} \left[\ell \left(f(\boldsymbol{x}; \boldsymbol{\theta}), y \right) \right] = \mathbb{E}_{\mathcal{D}_{\text{train}}} \left[-\boldsymbol{e}^y \log f \left(\boldsymbol{x}; \boldsymbol{\theta} \right) \right], \tag{6.2}$$

For the auxiliary dataset, we enforce the outputs to be consistent with dynamic noise labels, uniformly sampled from the label set in each epoch. Therefore, the training objective function is as follows:

$$\mathcal{L}_2 = \mathbb{E}_{\mathcal{D}_{\text{out}}} \left[\ell \left(f(\widetilde{\boldsymbol{x}}; \boldsymbol{\theta}), \widetilde{y} \right) \right] = \mathbb{E}_{\mathcal{D}_{\text{out}}} \left[-\boldsymbol{e}^{\widetilde{y}} \log f \left(\widetilde{\boldsymbol{x}}; \boldsymbol{\theta} \right) \right], \text{where } \widetilde{y} \sim \mathcal{U}_k \tag{6.3}$$

where $\mathcal{U}_k$ represents a discrete uniform distribution on the label set $\{1, \ldots, k\}$.

Combining the original training dataset and the auxiliary dataset in the training, now we can formalize ODNL as minimizing the final objective:

$$\mathcal{L}_{total} = \mathcal{L}_1 + \eta \cdot \mathcal{L}_2 = \mathbb{E}_{\mathcal{D}_{\text{train}}} \left[\ell \left(f(\boldsymbol{x}; \boldsymbol{\theta}), y \right) \right] + \eta \cdot \mathbb{E}_{\mathcal{D}_{\text{out}}} \left[\ell \left(f(\widetilde{\boldsymbol{x}}; \boldsymbol{\theta}), \widetilde{y} \right) \right] \tag{6.4}$$

where η denotes the trade-off hyperparameter. The details of ODNL are provided in Algorithm 3.

Extensions to Other Robust Training Algorithms. It is worthy to note that our regularization is a general method and can be easily incorporated into existing robust training algorithms, including sample selection[27, 59], loss correction [67], robust loss function [74, 76], etc. Given the original learning objective $\mathcal{L}_{\text{robust}}$ of the robust methods and the hyperparameter η, we formalize the final objective as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{robust}} + \eta \cdot \mathcal{L}_2 \tag{6.5}$$

Algorithm 3: Open-set Regularization with Dynamic Noisy Labels

Input: Training dataset $\mathcal{D}_{\text{train}}$. Open-set auxiliary dataset $\mathcal{D}_{\text{out}}$;

1: **for** each iteration **do**
2: Sample a mini-batch of original training data $\{(\boldsymbol{x}_i, y_i)\}_{i=0}^{n}$ from $\mathcal{D}_{\text{train}}$;
3: Sample a mini-batch of open-set data $\{\widetilde{\boldsymbol{x}}_i\}_{i=0}^{m}$ from $\mathcal{D}_{\text{out}}$;
4: Generate random noisy label $\widetilde{y}_i \sim \mathcal{U}_k$ for each open-set data $\widetilde{\boldsymbol{x}}_i$;
5: Perform gradient descent on f with $\mathcal{L}_{\text{total}}$ from Equation (6.4);
6: **end for**

6.4 Understanding from SGD Noise

To further understand the effect of our regularization, we provide an analysis from the lens of SGD noise. Following SLN [224], here we consider an original training sample $(\boldsymbol{x}, y)$ and an open-set sample $\widetilde{\boldsymbol{x}}$ at each SGD step for convenience. In parameter updates, our regularization adds noises to the original gradients with the open-set noisy sample $\widetilde{\boldsymbol{x}}$ and dynamic noisy label $\widetilde{y} \sim \mathcal{U}_k$:

$$\widetilde{\nabla}_{\boldsymbol{\theta}} \ell_{\text{total}} = \nabla_{\boldsymbol{\theta}} \ell(f(\boldsymbol{x}; \boldsymbol{\theta}), y) + \eta \cdot \nabla_{\boldsymbol{\theta}} \ell(f(\widetilde{\boldsymbol{x}}; \boldsymbol{\theta}), \widetilde{y}) \tag{6.6}$$

where ℓ denotes the cross entropy loss for single sample. For convenience, we omit the hyperparameter η and use $\boldsymbol{f}, \widetilde{\boldsymbol{f}}$ to indicate the model output $f(\boldsymbol{x}; \boldsymbol{\theta})$

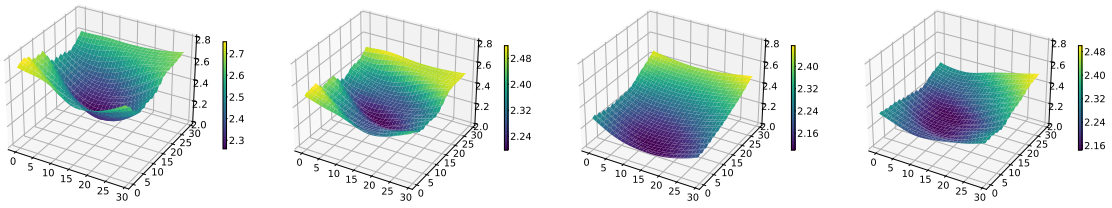
for the original training sample and the model output $f(\widetilde{\boldsymbol{x}};\boldsymbol{\theta})$ for the open-set noisy sample. Following the standard notation of the Jacobian matrix, we have $\nabla_{\boldsymbol{\theta}}\ell \in \mathbb{R}^{1\times \dot{p}}, \nabla_{\boldsymbol{f}}\ell \in \mathbb{R}^{1\times k}, \nabla_{\boldsymbol{\theta}}\boldsymbol{f} \in \mathbb{R}^{k\times p}, \nabla_{\boldsymbol{\theta}_i}\boldsymbol{f} \in \mathbb{R}^{k}$ and $\nabla_{\boldsymbol{\theta}}\boldsymbol{f}_i \in \mathbb{R}^{1\times p}$. The proofs of the following propositions are presented in Appendix C.

Proposition 6.1. *For the cross-entropy loss, Eq. (6.6) induces noise* $\boldsymbol{z} = -\frac{\nabla_{\boldsymbol{\theta}}\widetilde{\boldsymbol{f}}_j}{\widetilde{\boldsymbol{f}}_j}$ *on* $\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{f}, y)$*, s.t.,* $\boldsymbol{z} \in \mathbb{R}^p, j \sim \mathcal{U}_k$*, where* $\frac{\cdot}{\widetilde{\boldsymbol{f}}_j}$ *denotes the element-wise division. Note that the expected value of noise on the* i*-th parameter* $\boldsymbol{\theta}_i$ *is* $-\frac{1}{k}\sum_{j=1}^{k}\frac{\nabla_{\boldsymbol{\theta}_i}\widetilde{\boldsymbol{f}}_j}{\widetilde{\boldsymbol{f}}_j}$.

The effects of SGD noise. From Proposition 6.1, we find that our regularization induces SGD noise sampled from a uniform distribution over $\left\{-\nabla_{\boldsymbol{\theta}}\widetilde{\boldsymbol{f}}_j/\widetilde{\boldsymbol{f}}_j\right\}_{j=1}^{k}$. The effects of the noises can be analyzed from three aspects:

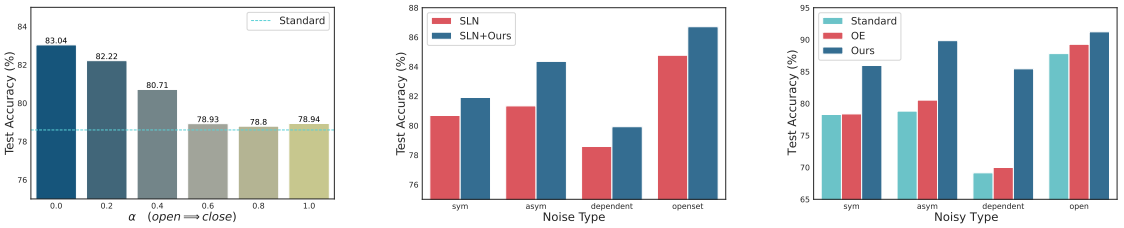
- Firstly, our regularization yields SGD noises with random direction, uniformly sampled from the gradients of the open-set noisy sample. In this way, the SGD noises make the training difficult to converge and help escape local minima [229–233]. Without the noises, the model would quickly converge to a local minimum because the optimization of parameters always follows the direction of gradient descent during the training.
- Secondly, the noises z are independently random to the gradients $\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{f}, y)$ from the original training example as the noises are conducted with the open-set sample $\widetilde{\boldsymbol{x}}$ from $\mathcal{D}_{out}$. If closed-set samples are used here, the generated noises might be conflicted with the original gradients, leading to unstable training and limiting the improvement from noises. This argument is clearly supported by the experimental results in Fig. 6.1 and Fig. 6.3a.
- Finally, the expected value of noise is not zero so the noise is biased. In particular, the expected value of noise enforces the model f to produce more conservative predictions on Out-of-Distribution samples. Consequently, this feature would lead to performance improvement on OOD detection tasks as shown in Subsection 6.5.4.

To further demonstrate the importance of open-set samples in the auxiliary dataset, we conduct experiments with synthetic auxiliary datasets consisting of convex combinations $\boldsymbol{x}'$ of open-set samples $\widetilde{\boldsymbol{x}}$ and closed-set samples $\boldsymbol{x}$ with a hyperparameter



(A) CE (B) OE (C) SLN (D) Ours

FIGURE 6.2: Loss landscapes around the local minimum of models trained on CIFAR-10 with symmetric-40% noisy labels. We compare their sharpness with the same-scale z-axis and show the loss distribution by drawing color bars separately. We show that our method and SLN lead to a flat minimum, while the models trained by Standard and OE converge to a sharp minimum.



(A) The importance of open set

(B) Our method can improve SLN

(C) Compare with OE.

FIGURE 6.3: (a) Test accuracy (%) of ODNL ($\eta = 1$) on CIFAR-10 with symmetric-40% inherent noisy labels using different synthetic auxiliary datasets: the larger the α is, the closer the auxiliary sample distribution $\mathcal{D}_{\text{out}}$ is to the origin sample distribution $\mathcal{D}_{\text{train}}$. (b) Comparison of performances on CIFAR-10 with different types of inherent noisy labels using SLN and SLN+Ours. Here we follow the settings of SLN. (c) Comparison of performances on CIFAR10 with different types of inherent noisy labels using OE and Ours.

α: $\boldsymbol{x}' = (1-\alpha) \cdot \widetilde{\boldsymbol{x}} + \alpha \cdot \boldsymbol{x}$. Here, we use 80 Million Tiny Images [226] and CIFAR-5m [227, 228] as the open-set and closed-set dataset. The results in Figure. 6.3a show that using open-set auxiliary dataset achieves better performance on robustness against inherent noises.

Relations to label randomization methods. Label randomization is also applied in some existing work in the context of deep learning[79, 224, 234]. For example, DisturbLabel [234] intentionally generates incorrect labels on a small fraction of training data to improve generalization performance under the settings of clean labels, interpreted as an implicit model ensemble. It can be seen as using a part of training data as a closed-set auxiliary dataset. In such a way, DisturbLabel can be interpreted as a special case of our method with a closed-set auxiliary dataset.

To mitigate the issue of label noise, Stochastic Label Noise (SLN) [224] introduces a variant of SGD noise induced by adding dynamic and mean-zero perturbations on the labels of training dataset, while Label Smoothing [79] uses a fixed and biased perturbation on the label. The noise gradients induced by SLN are given as follows:

$$\tilde{\nabla}_{\boldsymbol{\theta}}\ell(\boldsymbol{f}, y) = \nabla_{\boldsymbol{\theta}}\ell\left(\boldsymbol{f}, \boldsymbol{e}^{y} + \sigma_y \boldsymbol{z}_y\right) \tag{6.7}$$

where $\sigma_y > 0, \boldsymbol{z}_y \in \mathcal{R}^k$, $\boldsymbol{z}_y \sim \mathcal{N}\left(0, \boldsymbol{I}_{k\times k}\right)$.

Proposition 6.2 (Proposition 3 in Chen et al. [224])**.** *For the cross-entropy loss, Eq. (6.7) in SLN induces noise* $\boldsymbol{z} \sim \mathcal{N}(0, \sigma_y^2 M)$ *on* $\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{f}, y)$*, s.t.,* $\boldsymbol{M} \in \mathbb{R}^{p\times p}$ *and* $\boldsymbol{M}_{i,j} = \left(\frac{\nabla_{\boldsymbol{\theta}_i}\boldsymbol{f}}{\boldsymbol{f}}\right)^T \frac{\nabla_{\boldsymbol{\theta}_j}\boldsymbol{f}}{\boldsymbol{f}}, \forall i, j \in \{1, \ldots, p\}$*, where* $\frac{\cdot}{\boldsymbol{f}_j}$ *denotes the element-wise division.*

From Proposition 6.2, we show that SLN induces SGD noises that obey a Gaussian distribution, where the standard deviation can be adjusted by the hyperparameter σ_y. Therefore, SLN and our ODNL can be considered as different variants of SGD noise, helping the network converge to a flat minimum, which is expected to provide better generalization performance [229–233]. The effects are explicitly presented in Fig. 6.2c and Fig. 6.2d. Moreover, the results in Fig. 6.3b illustrate that our method could further improve the performance of SLN[3] under various types of label noise, showing the complementarity between SLN and our method.

Relations to Outlier Exposure. Outlier Exposure (OE) [3] is an effective method designed for OOD detection tasks. Specifically, OE also makes use of a large, unlabeled auxiliary dataset sampled from $\mathcal{D}_{\text{out}}$, and regularizes the softmax probabilities to be a uniform distribution for out-of-distribution data in the auxiliary dataset. In such manner, OE train the model to discover signals and learn heuristics to detect whether a query is sampled from $\mathcal{D}_{\text{train}}$ or $\mathcal{D}_{\text{out}}$. The training objective is to minimize: $\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{\text{train}}}\left[\ell(f(\boldsymbol{x};\boldsymbol{\theta}), y)\right] + \lambda \cdot \mathbb{E}_{\widetilde{\boldsymbol{x}}\sim\mathcal{D}_{\text{out}}}\left[-\frac{1}{k}\cdot\sum_{i=1}^{k}\log f_i\left(\widetilde{\boldsymbol{x}};\boldsymbol{\theta}\right)\right]$.

Proposition 6.3. *For the cross-entropy loss, each open-set sample in OE induces bias* $\boldsymbol{z} = -\frac{1}{k}\sum_{j=1}^{k}\frac{\nabla_{\boldsymbol{\theta}}\widetilde{\boldsymbol{f}}_j}{\widetilde{\boldsymbol{f}}_j}$ *on* $\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{f}, y)$*, s.t.,* $\boldsymbol{z} \in \mathbb{R}^p$*, where* $\frac{\cdot}{\widetilde{\boldsymbol{f}}_j}$ *denotes the element-wise division.*

Compared Proposition 6.3 to Proposition 6.1, our method should have similar regularization effect to OE on OOD instances, because the expectation of the SGD

[3]We use the official implementation: `https://github.com/chenpf1025/SLN`.

TABLE 6.1: Average test accuracy (%) with standard deviation on CIFAR-10 under various types of noisy labels (over 5 trials). The bold indicates the improved results by integrating our regularization.

Method	Sym-20%	Sym-50%	Asymmetric	Dependent	Open-set
Standard	86.93±0.49	69.48±0.58	78.34±0.96	68.39±0.65	88.45±0.57
+ **ODNL (Ours)**	**89.94±0.70**	**81.20±0.68**	**86.91±0.32**	**82.42±0.36**	**91.12±0.39**
Decoupling	88.31±0.39	80.81±0.85	84.85±0.22	82.44±0.54	84.69±0.46
+ **ODNL (Ours)**	**89.27±0.90**	**81.33±0.77**	**87.03±0.76**	**84.14±0.86**	**86.34±0.32**
F-correction	84.95±0.27	72.11±0.15	84.46±0.96	72.44±0.58	85.54±0.86
+ **ODNL (Ours)**	**87.95±0.28**	**82.44±0.21**	**87.56±0.19**	**81.83±0.43**	**89.67±0.69**
PHuber-CE	86.69±0.31	70.05±0.55	78.10±0.44	70.36±1.12	88.09±0.14
+ **ODNL (Ours)**	**91.98±0.38**	**83.19±0.26**	**89.18±0.46**	**80.87±0.48**	**91.62±0.15**
Co-teaching	91.72±0.35	86.65±0.43	87.56±0.36	88.64±0.79	89.91±0.58
+ **ODNL (Ours)**	**92.80±0.75**	**89.64±0.81**	**88.39±0.39**	**91.36±0.23**	**92.60±0.47**
JoCoR ($\lambda = 0.5$)	92.68±0.33	88.36±0.64	84.06±0.37	90.11±0.19	88.80±0.54
+ **ODNL (Ours)**	**93.41±0.75**	**90.40±0.43**	**86.46±0.80**	**91.75±0.92**	**91.62±0.88**

noises from ODNL is equal to the bias induced by OE. However, OE cannot improve the robustness against noisy labels as it does not induce dynamic noises so that the optimization of parameters always follows the direction of gradient descent. The difference is empirically shown in Fig. 6.3c and we further verify it in Subsection 6.5.4.

6.5 Experiments

In this section, we verify the effectiveness of our method on three benchmark datasets: CIFAR-10/100 and Clothing1M. We show that ODNL could not only significantly improve the robustness against various types of inherent noisy labels in standard training, but also enhance the performance of existing robust training techniques. Then we perform a sensitivity analysis to validate the effect of η. Finally, we conduct experiments to evaluate the performance of ODNL on OOD detection task.

6.5.1 Setups

We comprehensively verify the utility of ODNL on different types of label noise, including symmetric noise, asymmetric noise[76], instance-dependent noise [68] and

open-set noise [33] synthesized on CIFAR-10/100 and real-world noise on Clothing1M [155]. Symmetric noise assumes each label has the same probability of flipping to any other classes. We uniformly flip the label to other classes with an overall probability 20% and 50%. Asymmetric noise assumes labels might be only flipped to similar classes [67, 76, 224]. Noisy labels are generated by mapping TRUCK $\rightarrow$ AUTOMOBILE, BIRD $\rightarrow$ AIRPLANE, DEER $\rightarrow$ HORSE, and CAT $\leftrightarrow$ DOG with probability 40% for CIFAR-10. For CIFAR-100, we flip each class into the next circularly with probability 40%. Instance-dependent noise assumes the mislabeling probability is dependent on each instance's input features [68, 224, 235]. We use the instance-dependent noise from PDN [235] with a noisy rate 40%, where the noise is synthesized based on the DNN prediction error. Open-set noise contains samples that do not belong to the label set considered in the classification task [33]. We generate open-set noises on CIFAR-10 by randomly replacing 40% of its training images with images from CIFAR-100.

We perform training with WRN-40-2 [1] on CIFAR-10 and CIFAR-100. The network is trained for 200 epochs using SGD with a momentum of 0.9, a weight decay of 0.0005. In each iteration, both the batch sizes of the original dataset and the open-set auxiliary dataset are set as 128. We set the initial learning rate as 0.1, and reduce it by a factor of 10 after 80 and 140 epochs. We use 5k noisy samples as the validation to tune the hyperparameter η in $\{0.1, 0.5, 1, 2.5, 5\}$, then train the model on the full training set and report the average test accuracy in the last 5 epochs. All experiments are repeated five times with different seeds. We use 80 Million Tiny Images [226] as the open-set auxiliary dataset.

6.5.2 Results on CIFAR-10 and CIFAR-100

On CIFAR-10 and CIFAR-100, we verify that ODNL can boost existing robust training methods by integrating ODNL with the following methods: 1) Decoupling [28], which updates the parameters only using instances that have different predictions from two classifiers. 2) F-correction [67], which corrects the prediction by the label transition matrix. As the setting in F-correction, we first train a standard network to estimate the transition matrix Q. 3) PHuber-CE [78], which uses a composite loss-based gradient clipping, a variation of standard gradient clipping

TABLE 6.2: Average test accuracy (%) with standard deviation on CIFAR-100 under various types of noisy labels (over 5 trials). The bold indicates the improved results by integrating our regularization.

Method	Sym-20%	Sym-50%	Asymmetric	Dependent
Standard	64.87±0.81	48.55±0.65	48.77±0.86	53.94±0.52
+ **ODNL (Ours)**	**68.98±0.16**	**54.35±0.63**	**55.07±0.38**	**60.47±0.27**
Decoupling	64.26±0.23	49.09±0.81	49.92±0.71	55.17±0.34
+ **ODNL (Ours)**	**67.06±0.25**	**53.65±0.56**	**57.74±0.54**	**59.32±0.39**
F-correction	62.05±0.35	52.27±0.15	54.45±0.94	57.47±0.26
+ **ODNL (Ours)**	**65.60±0.21**	**57.34±0.37**	**57.60±0.52**	**61.83±0.93**
PHuber-CE	65.17±0.87	48.46±0.24	47.64±0.45	53.56±0.29
+ **ODNL (Ours)**	**72.18±0.41**	**63.02±0.46**	**49.27±0.38**	**67.16±0.25**
Co-teaching	72.35±0.60	65.21±0.88	54.16±0.17	67.85±0.82
+ **ODNL (Ours)**	**73.34±0.13**	**65.43±0.61**	**55.02±0.29**	**68.41±0.14**
JoCoR ($\lambda = 0.5$)	73.44±0.19	67.53±0.79	57.02±0.32	70.01±0.26
+ **ODNL (Ours)**	**74.74±0.21**	**68.17±0.34**	**58.57±0.46**	**71.68±0.40**

for label noise robustness. 4) Co-teaching [27], which trains two networks simultaneously and cross-updates parameters of peer networks. 5) JoCoR [59], which trains two networks simultaneously and reduces the divergence between the two classifiers by explicit regularization.

The average test accuracy with WRN-40-2 [1] are reported in Table 6.1 and Table 6.2. The results show that incorporating our regularization into existing robust training methods consistently improves their performance against various types of noisy labels. In particular, we can observe that our method can boost almost all types of existing methods, including robust loss function, sample selection, loss correction, etc., showing that our regularization based upon an open-set auxiliary dataset is a complementary method to existing robust algorithms. More intriguingly, while most existing methods do not bring benefits to the robustness against open-set noisy labels in the training set of CIFAR-10, our regularization could still enhance the performance of all existing methods in this case.

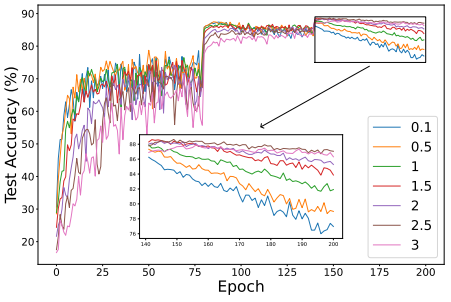


FIGURE 6.4: Results of sensitivity analysis on CIFAR-10 with different η.

To further illustrate the influence of η, we present a sensitivity analysis on CIFAR-10 with symmetric-40% noisy labels in Fig. 6.4. Specifically, we highlight the differences in the trend of test accuracy after the second decay of learning rate at

TABLE 6.3: Test accuracy (%) on Clothing1M with pretrained ResNet-50. All experiments are implemented based on DivideMix's official implementation. The bold indicates the improved results.

Method	Standard	Forward	Co-teaching	**ODNL (Ours)**	SLN	**+ ODNL (Ours)**	DivideMix	**+ ODNL (Ours)**
best	70.17	70.46	71.32	**72.47**	72.31	**73.12**	74.32	**74.89**
last	68.23	69.34	71.02	**72.38**	71.94	**72.98**	73.83	**74.35**

TABLE 6.4: OOD detection performance comparison on CIFAR-10 with symmetric-40% noisy labels. All values are percentages and are averaged over the ten test datasets. ↑ indicates larger values are better and ↓ indicates smaller values are better. Bold numbers are superior results.

Method	Test Accuracy ↑	FPR95 ↓	AUROC ↑	AUPR ↑
MSP	77.87	92.01	54.64	19.41
MSP+OE	78.12	26.76	94.02	78.34
MSP+Ours	**85.31**	**14.08**	**96.53**	**84.10**

the 140th epoch. We can observe that with a large η, the decreasing trend of test accuracy caused by inherent noisy labels is nearly eliminated. The result verifies that our regularization is an effective method to reduce the overfitting of inherent noisy labels.

6.5.3 Results on Clothing1M

Clothing1M [155] is a large-scale dataset with real-world noisy labels of clothing images. There are 1 million noisy samples for training, $14k$ and $10k$ clean examples for validation and test respectively. Following DivideMix [62], we conduct experiments by sampling a class-balanced training subset in each epoch and use ResNet-50 with ImageNet pretrained weights. We set the η of the vanilla ODNL as 5. For the auxiliary dataset in our method, we use ImageNet 2012 dataset [85] and remove all classes related to clothes. Other training details strictly follow the settings of DivideMix [62]. As shown in Table 6.3, *best* denotes the score of the epoch where the validation accuracy is optimal, and *last* denotes the scores at the last epoch. The vanilla ODNL outperforms many simple baselines, and it can further improve the performance of SLN and DivideMix. The results show that our method is applicable for large-scale real-world scenarios.

6.5.4 OOD detection with noisy labels

Out-of-distribution (OOD) detection is an essential problem for the deployment of deep learning especially in safety-critical applications [3, 31]. From the analysis in Section 6.4, we know that our regularization should have a similar regularization effect to OE [3], which enforces the model to give conservative outputs on OOD instances. Here, we conduct experiments on CIFAR-10 with symmetric-40% noisy labels to verify the advantage of our method on OOD detection tasks even in a weakly supervised setting. We expect the trained models could perform well in the OOD detection task, even if the labels of training dataset are noisy. The training settings are the same as those in subsection 6.5.2. Following OE [3], we consider the maximum softmax probability (MSP) baseline [115], which gives the maximum of softmax output as the OOD score. For the OOD test datasets, we use three types of noises and seven common benchmark datasets: Gaussian, Rademacher [3], Blobs [3], Textures [195], SVHN[196], CIFAR100, Places365[86], LSUN-Crop [197], LSUN-Resize [197], and iSUN [198]. We measure the following metrics: (1) the false positive rate (FPR95) of OOD instances when the true positive rate of in-distribution examples is at 95%; (2) the area under the receiver operating characteristic curve (AUROC); and (3) the area under the precision-recall curve (AUPR). Table 6.4 presents the test accuracy and the average performance over the three types of noises and seven OOD test datasets. We can observe that our method achieves impressive improvement on both the test accuracy and the detection performance, while OE does not bring benefit on the generalization performance. The results can serve as evidence for the analysis in Section 6.4.

6.6 Chapter Summary

In this chapter, we proposed Open-set regularization with Dynamic Noisy Labels (ODNL), a simple yet effective technique that enhances many existing robust training methods to mitigate the issues of inherent noisy labels across various settings. To the best of our knowledge, our method is the first to utilize open-set auxiliary dataset in the problem of noisy labels. We show that ODNL can be considered as a variant of SGD noises that usually leads to a flat minimum. Extensive experiments

show that ODNL can help improve the robustness against various types of noisy labels and it is applicable for large-scale real-world scenarios. Moreover, ODNL also brings a "side effect": it can improve the OOD detection performance, which is also essential for the deployment of deep learning in safety-critical applications. On the other hand, ODNL is computationally inexpensive and can be easily integrated into existing algorithms. Overall, our method is an effective and complementary approach for boosting robustness against inherent noisy labels.

Chapter 7

Rebalancing Long-tailed Datasets with OOD Instances

7.1 Motivation

The success of deep neural networks (DNNs) heavily relies on large-scale datasets with balanced distribution [84, 85]. However, in real-world applications like autonomous driving and medical diagnosis, large-scale datasets naturally exhibit imbalanced and long-tailed distributions, i.e., a few classes (majority classes) occupy most of the data while most classes (minority classes) are under-represented [86–88]. It has been shown that training on long-tailed datasets leads to poor generalization performance, especially on the minority classes [89–91]. Thus, designing effective algorithms to handle class imbalance is of great practical importance.

In the literature, a popular direction in long-tailed learning is to re-balance the data distribution by data re-sampling [93, 94]. For example, Over-sampling [96, 97] repeats samples from under-presented classes, but it usually causes over-fitting to the minority classes. To alleviate the over-fitting issue, synthesized novel samples are introduced to augment the minority classes without repetition [98]. Nevertheless, the model is still error-prone due to noise in the synthesized samples [102]. A recent work [112] introduced unlabeled-in-distribution data to compensate for the lack of training samples, and showed that directly adding unlabeled data from mismatched classes (*i.e.*, out-of-distribution data) by semi-supervised learning hurts

the generalization performance. These data augmentation methods normally require in-distribution data with precise labels for selected classes. However, such kind of data would be extremely hard to collect in real-world scenarios, due to the expensive labeling cost. This fatal weakness of previous methods motivates us to explore the possibility of using *out-of-distribution* (OOD) data for long-tailed imbalanced learning.

In this chapter, we theoretically show that out-of-distribution data (i.e., open-set samples) could be leveraged to augment the minority classes from a Bayesian perspective. Based on this motivation, we propose a simple yet effective method called Open-sampling [1], which uses open-set noisy labels to re-balance the label priors of the training dataset. For each OOD instance, the label is sampled from our pre-defined distribution that is complementary to the original class priors. To alleviate the over-fitting issue on the minority classes, a class-dependent weight is used in the training objective to provide stronger regularization on the minority classes than the majority classes. In this way, the open-set noisy labels could be used to re-balance the class priors while retaining their non-toxicity.

To provide a comprehensive understanding, we conduct a series of analyses to illustrate the properties of the proposed Open-sampling method. From these empirical analyses, we show that: 1) the Complementary Distribution is superior to the commonly used Class Balanced distribution (CB) as the former is closer to the uniform distribution, which reduces the harmfulness of the open-set noisy labels; 2) real-world datasets with large sample size are the best choices for the open-set auxiliary dataset in Open-sampling and the diversity (i.e., number of classes) is not an important factor in the method; 3) the Open-sampling method not only re-balances the class prior, but also promotes the neural network to learn more separable representations.

To the best of our knowledge, we are the first to explore the benefits of OOD instances in learning from long-tailed datasets. To verify the effectiveness of our method, we conduct experiments on four long-tailed image classification benchmark datasets, including long-tailed CIFAR10/100 [84], CelebA-5 [6, 237], and Places-LT [86]. Empirical results show that our method can be easily incorporated into existing state-of-the art methods to enhance their performance on long-tailed imbalanced classification tasks. Furthermore, experimental results validate

[1] The work in this chapter has been published in [236].

that our method could also achieve impressive performance for detecting OOD instances under class-imbalanced setting. Code and data are publicly available at `https://github.com/hongxin001/open-sampling`.

7.2 Preliminaries

7.2.1 Background

In this work, we consider a multi-class classification problem, where the input space is denoted by $\mathcal{X} \in \mathbb{R}^d$ and the label space $\mathcal{Y}$ is $\{1, \ldots, K\}$. We denote by $\mathcal{D}_{\text{train}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N \in \mathcal{X} \times \mathcal{Y}$ the training dataset with N samples. Let n_j be the number of samples in class j, then $N = \sum_{j=1}^K n_j$. Let $P_{\text{s}}(X, Y)$ define the underlying training (source) distribution and $P_{\text{t}}(X, Y)$ define the test (target) distribution. Generally, the class imbalance problem assumes that the test data has the same class conditional probability as the training data, i.e., $P_{\text{s}}(X|Y) = P_{\text{t}}(X|Y)$, while their class priors are different, i.e., $P_{\text{s}}(Y) \neq P_{\text{t}}(Y)$.

Besides, we consider an unlabelled auxiliary dataset $\mathcal{D}_{\text{out}}^{(\boldsymbol{x})} = \{\tilde{\boldsymbol{x}}_i\}_{i=1}^M \in \mathcal{X}$ consisting of M open-set instances, and we have $M \gg N$. These open-set instances are also known as OOD data as they are sampled from $P_{\text{out}}(X)$, which is disjoint from $P_{\text{s}}(X)$. In real-world scenarios, it is easy to obtain such auxiliary datasets, which are commonly used in OOD detection tasks. In what follows, we may assign each open-set instance $\tilde{\boldsymbol{x}}_i$ with a random label $\tilde{y}_i \in \mathcal{Y}$, which is independently sampled from an appropriate label distribution $P_{\text{out}}(Y)$ over $\mathcal{Y}$. We denote by $\mathcal{D}_{\text{out}} = \{(\tilde{\boldsymbol{x}}_i, \tilde{y}_i)\}_{i=1}^M$ the auxiliary dataset with randomly sampled noisy labels.

7.2.2 Theoretical motivation

From a Bayesian perspective, the prediction of a Bayes classifier is generally made as follows:

$$y^* = \arg\max_{y \in \mathcal{Y}} P(y|\boldsymbol{x}) = \arg\max_{y \in \mathcal{Y}} P(\boldsymbol{x}|y)P(y), \tag{7.1}$$

where $P(\boldsymbol{x})$ and $P(y)$ represent $P(X = \boldsymbol{x})$ and $P(Y = y)$, respectively. Unfortunately, the predicted posterior probability in Eq. (7.1) becomes unreliable when

there is a large discrepancy between the class priors of training and test distributions. Specifically, the class prior of the test distribution is usually balanced (i.e., a uniform distribution over labels), while the training dataset exhibits a long-tailed class distribution. Formally, we have $P_{\mathrm{s}}(Y=i) \neq P_{\mathrm{t}}(Y=i)$ for any class $i \in \mathcal{Y}$, and $P_{\mathrm{s}}(Y=j) \ll P_{\mathrm{t}}(Y=j)$ for a minority class $j \in \mathcal{Y}$.

To re-balance the class priors of the training dataset, existing re-sampling methods mostly augment the minority classes with extra *in-distribution* samples with precise labels, e.g., duplicated examples [95], synthetic generation [99], and interpolation [98]. However, due to their strict in-distribution constraints on data distribution and label quality, generating and collecting those samples are challenging and expensive, especially for minority classes. In this chapter, we instead try to break through the constraints by demonstrating that OOD instances with noisy labels are actually useful for re-balancing the training dataset.

We start by presenting an intriguing fact in the following theorem (proof in Appendix D.1), which demonstrates that augmenting the training data with open-set instances and uniformly random labels can be non-toxic.

Theorem 7.1. *Assume that $P_{\mathrm{out}}(Y)$ is the discrete uniform distribution over the label space $\mathcal{Y}$. Let the augmented dataset be $\mathcal{D}_{\mathrm{mix}} = \mathcal{D}_{\mathrm{train}} \cup \mathcal{D}_{\mathrm{out}}$, and $P_{\mathrm{mix}}(X, Y)$ be the underlying data distribution of $\mathcal{D}_{\mathrm{mix}}$, then we have*

$$\arg\max_{y \in \mathcal{Y}} P_{\mathrm{mix}}(\boldsymbol{x}|y) P_{\mathrm{mix}}(y) = \arg\max_{y \in \mathcal{Y}} P_{\mathrm{s}}(\boldsymbol{x}|y) P_{\mathrm{s}}(y).$$

Theorem 7.1 indicates that, when the labels are uniformly sampled from the in-distribution label space, the prediction of the Bayes classifier is unchanged after augmenting the training dataset with the OOD instances[2]. In this way, the theoretical result motivates us to exploit the potential value of OOD instances to repair class imbalance.

Although using the uniform distribution as $P_{\mathrm{out}}(Y)$ will not downgrade the Bayes classifier as shown above, the resulting classifier is not yet optimal, and its corresponding partition is still far away from the optimal decision boundary on the test distribution, since the label distribution $P_{\mathrm{mix}}(Y)$ still remains largely imbalanced.

[2]The discovery of the non-toxicity of OOD instances is not new and has been studied in ODNL [238]. Our unique contribution here is to theoretically prove this property from a Bayesian perspective. We provide a detailed comparison for these two works in Subsection 7.3.

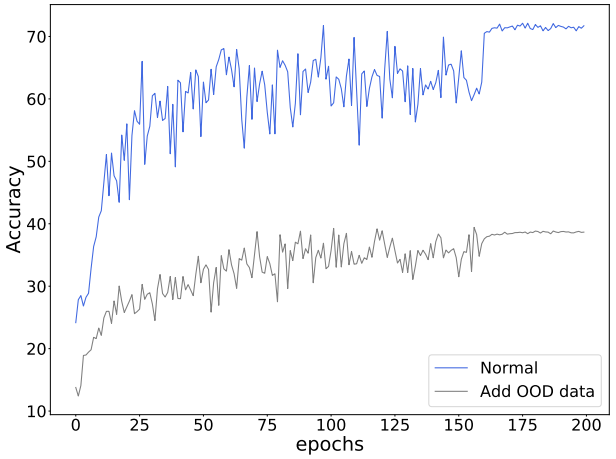


FIGURE 7.1: The test accuracy of models under different training epochs. Blue color denotes the model trained on long-tailed CIFAR-10 with ResNet-34, and gray color denotes the model trained on long-tailed CIFAR-10 and OOD data. We use instances from 300K Random Images [3] as OOD data and label as a minority class—"9". The comparison implies that simply adding OOD data into training may downgrade the generalization performance.

In the following, we will show that a better classifier can be obtained by exploring the trade-off between re-balancing the label distribution and keeping non-toxicity.

7.3 The Proposed Approach

Motivated by the previous analysis, we propose to exploit the open-set auxiliary dataset to improve the generalization under class-imbalanced settings. With a proper label distribution, OOD instances with dynamic labels can be used to re-balance the class priors while retaining their non-toxicity. We start by giving the following definition.

Definition 7.1 (Complementary Distribution)**.** Complementary Distribution (CD) is a label distribution for the auxiliary dataset to re-balance the class priors of the original dataset. In particular, Minimum Complementary Distribution (MCD) is the complementary distribution that requires the smallest number of auxiliary instances to re-balance the original training set.

Designing a proper Complementary Distribution to mitigate the class imbalance problem is a difficult problem that depends on the trade-off between re-balancing the class priors and keeping the non-toxicity of the added noisy labels. Intuitively, to re-balance the class priors, more OOD instances should be allocated into the

minority classes than the majority classes. Meanwhile, the unequal number of OOD instances in different classes may shift the Bayes classifier.

The above conflict can be understood naturally through a simple case. We consider a binary classification task with $K = 2$, and let the sample numbers of the two classes be n_1 and n_2, respectively. Without loss of generality, let $n_1 > n_2$. It is straightforward to verify that one of the optimal allocation to re-balance the class priors is simply appending $n_1 - n_2$ extra OOD instances into the minority class. However, in such a way, all OOD instances are assigned the same label, thereby downgrading the resulting classifier.

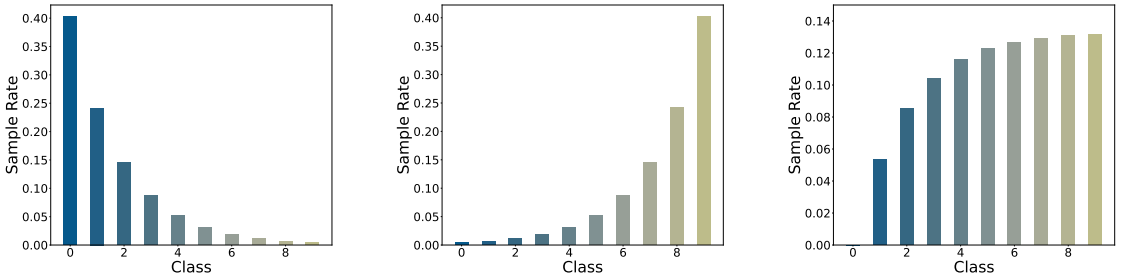
To provide a straightforward view, we show in Figure 7.1 the performance of model trained with the extra OOD data. Indeed, in the extreme case where all OOD instances are assigned the same label, their toxicity would be enlarged during training, leading to poor generalization performance.

Complementary Sampling Rate. To find a "sweet spot", we propose the following sampling rate that allows us to achieve a smooth transition from the MCD to the uniform distribution. We denote CD as Γ, MCD as Γ^m, and the complementary sampling rate for class j as Γ_j, then we have the following proposition.

Proposition 7.1 (Complementary Sampling Rate)**.** $\Gamma_j = (\alpha - \beta_j)/(K \cdot \alpha - 1)$, *where* $\beta_j = \frac{n_j}{\sum_{i=1}^{K} n_i}$. *Then, (i)* $\sum_{i=1}^{K} \Gamma_i = 1$*; (ii)* $\Gamma = \Gamma^m$ *if* $\alpha = \max_j(\beta_j)$*; (iii)* $\Gamma_j \to 1/K$ *as* $\alpha \to \infty$.

The hyperparameter $\alpha \in \mathbb{R}^+ \geq \max_j(\beta_j)$ controls the trade-off bewteen the MCD and uniform distribution. As shown in Proposition 7.1, when $\alpha = \max_j(\beta_j)$, it recovers the MCD. With a larger value for the α, the label distribution of the auxiliary dataset would be closer to a uniform distribution. The proof of Proposition 7.1 is provided in Appendix D.2.

As shown in Figure 7.2, both the CB and MCD distributions exhibit an inverse relationship with the original label prior, i.e., the minority classes possess larger sampling rate than the majority classes. Compared with the CB distribution, our MCD distribution is flatter so that the instances would not be concentrated in a class. In such a manner, the harmfulness of open-set noisy labels would be alleviated to a large extent. The advantage of the MCD distribution is empirically supported in Section 7.5.



(A) Original (B) CB (C) MCD

FIGURE 7.2: An illustration of label distributions for long-tailed CIFAR-10 dataset with imbalance ratio 100. (7.2a) Original label prior, (7.2b) Class Balanced distribution, (7.2c) Minimum Complementary Distribution.

With the Complementary Sampling Rate, we can then build an auxiliary dataset with OOD instances to re-balance the training dataset while retaining their non-toxicity.

Training Objective. To involve the sampled open-set noisy labels into the training, a natural idea is directly combining them with the original training dataset, and using the standard cross entropy as the training objective function:

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_{P_{\text{mix}}(X,Y)}[\ell(f(\boldsymbol{x};\boldsymbol{\theta},y)] \\ &= \mathbb{E}_{P_{\text{mix}}(X,Y)}[-\mathbf{e}^y \log f(\boldsymbol{x};\boldsymbol{\theta})],\end{aligned}$$

where $\boldsymbol{e}^y \in \{0,1\}^K$ denotes the one-hot vector whose y-th entry of $\boldsymbol{e}^y$ is 1. In each epoch, the labels of samples from the auxiliary dataset could be updated following the Complementary Distribution Γ.

However, the naive combination would consume too much capacity of the network on fitting the open-set noisy labels, making it hard to converge, especially when the sample size of the auxiliary dataset is much larger than that of the original training dataset. To handle this issue, we propose to use the loss on the auxiliary dataset as a regularization term as shown below:

$$\mathcal{L}_{\text{reg}} = \mathbb{E}_{\widetilde{\boldsymbol{x}}\sim P_{\text{out}}(X)}\left[\ell\left(f(\widetilde{\boldsymbol{x}};\boldsymbol{\theta}),\widetilde{y}\right)\right],$$

where $\widetilde{y}$ is drawn from the Complementary Distribution Γ.

To further alleviate the over-fitting issue on the minority classes without sacrificing the performance on the majority classes, we explicitly introduce a class-dependent weighting factor ω_j based on the pre-defined Complementary Distribution. To

make the total loss roughly in the same scale after applying ω_j, we normalize ω so that $\sum_{j=1}^{K} \omega_j = K$. Then, the regularization item becomes:

$$\mathcal{L}_{\text{reg}} = \mathbb{E}_{\widetilde{\boldsymbol{x}} \sim P_{\text{out}}(X)} \left[\omega_{\widetilde{y}} \cdot \ell \left(f(\widetilde{\boldsymbol{x}}; \boldsymbol{\theta}), \widetilde{y} \right) \right],$$

where $\widetilde{y} \sim \Gamma$ and $\omega_{\widetilde{y}} = \Gamma_{\widetilde{y}} \cdot K$. Now, the final training objective function is given as follows:

$$\begin{aligned} \mathcal{L}_{\text{total}} = & \mathbb{E}_{((\boldsymbol{x},y) \sim P_{\text{s}}(X,Y))} \left[\ell \left(f(\boldsymbol{x}; \boldsymbol{\theta}), y \right) \right] \\ & + \eta \cdot \mathbb{E}_{(\widetilde{\boldsymbol{x}}) \sim P_{\text{out}}(X)} \left[\omega_{\widetilde{y}} \cdot \ell \left(f(\widetilde{\boldsymbol{x}}; \boldsymbol{\theta}), \widetilde{y} \right) \right], \end{aligned} \tag{7.2}$$

where η controls the strength of the regularization term.

As a data re-balancing technique, Open-sampling is orthogonal to existing methods, including the training objective based methods (e.g., LDAM [110], Balanced Softmax [7]), Decoupled training methods [91], and self-supervised pre-trained methods [112]. Our method can be easily incorporated into these algorithms to further improve their generalization performance. Given the original learning objective $\mathcal{L}_{\text{imb}}$ of the existing methods, we can formalize the final objective as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{imb}} + \eta \cdot \mathcal{L}_{\text{reg}}. \tag{7.3}$$

Relation to ODNL. Recent work [238] shows that open-set noisy labels could be applied to enhance the robustness against inherent noisy labels, which has some conceptual similarities to the proposed method in this work. Here, we summarize the main differences between ODNL [238] and our work.

1. Problem setting: ODNL focuses on improving the robustness against noisy labels while our work considers the problem of learning from long-tailed imbalanced datasets.

2. Technique: we proposed to sample the labels of OOD instances from the Complementary Distribution and add a weight factor to their losses, while they treats all the OOD instances equally by simply using a uniform distribution. In particular, ODNL can be seen as a special case of our method with a large value of α. As analyzed in Figures 7.4a, 7.4b, and 7.4c, our

Open-sampling consistently outperforms the variant with a uniform distribution or a large value of α, which demonstrates the advantage of the proposed method.

3. Insight: ODNL aims to consume the extra representation capacity of neural networks to prevent over-fitting inherent noisy labels and show that their method helps the network converge to a flat minimum as SGD noises. In our work, OOD instances are applied to re-balance the label prior of the training dataset and the proposed method are shown to encourage the network to learn more separable representations.

7.4 Experiments

In this section, we evaluate our proposed method on long-tailed image classification datasets, including long-tailed CIFAR10/100 [84], CelebA-5 [6, 237], Places-LT [86]. Then we analyze the impact of η by sensitivity analysis. Finally, we conduct experiments to evaluate the performance of Open-sampling on OOD detection task under class-imbalanced setting.

7.4.1 Datasets

Long-Tailed CIFAR. The original version of CIFAR-10 and CIFAR-100 contains 50,000 training images and 10,000 validation images of size 32×32 with 10 and 100 classes, respectively. To create their long-tailed version, we reduce the number of training examples per class according to an exponential function $n = n_j \mu^j$, where j is the class index, n_j is the original number of training images, and $\mu \in (0, 1)$. Besides, the validation set and the test set are kept unchanged. The imbalance ratio of a dataset is defined as the number of training samples in the largest class divided by that of the smallest.

CelebA-5. CelebFaces Attributes (CelebA) dataset is a real-world long-tailed dataset. It is originally composed of 202,599 number of RGB face images with 40 binary attributes annotations per image. Note that CelebA is originally a multi-labeled dataset, we port it to a 5-way classification task by filtering only the samples with five non-overlapping labels about hair colors. We also subsampled the full

dataset by 1/20 while maintaining the imbalance ratio as 10.7, following [6]. In particular, We pick out 50 and 100 samples in each class for validation and testing. We denote the resulting dataset by CelebA-5.

Places-LT. Places-LT is a long-tailed version of the large-scale scene classification dataset Places [86]. It consists of 184.5K images from 365 categories with class cardinality ranging from 5 to 4,980.

Auxiliary datasets. 300K Random Images [3] is a cleaned and debiased dataset with 300K natural images. In this dataset, Images that belong to CIFAR classes from it, images that belong to Places or LSUN classes, and images with divisive metadata are removed so that $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{out}}$ are disjoint. We use the dataset as the open-set auxiliary dataset for experiments with CIFAR-10/100 and CelebA-5. For Experiments on Places-LT, we use the training set of Places-Extra69 [86] as the open-set auxiliary dataset.

OOD test datasets. Following OE [3], we comprehensively evaluate OOD detectors on artificial and real anomalous distributions, including: Gaussian, Rademacher, Blobs, Textures [195], SVHN [196], Places365 [86], LSUN-Crop [197], LSUN-Resize [197], iSUN [198]. For experiments on CIFAR-10, we also use CIFAR-100 as OOD test dataset. *Gaussian* noises have each dimension i.i.d. sampled from an isotropic Gaussian distribution. *Rademacher* noises are images where each dimension is -1 or 1 with equal probability, so each dimension is sampled from a symmetric Rademacher distribution. *Blobs* noises consist in algorithmically generated amorphous shapes with definite edges. *Textures* [195] is a dataset of describable textural images. *SVHN* dataset [196] contains 32×32 color images of house numbers. There are ten classes comprised of the digits 0-9. *Places365* [86] consists in images for scene recognition rather than object recognition. *LSUN* [197] is another scene understanding dataset with fewer classes than Places365. Here we use *LSUN-Crop* and *LSUN-Resize* to denote the cropped and resized version of the LSUN dataset respectively. *iSUN* [198] is a large-scale eye tracking dataset, selected from natural scene images of the SUN database [239].

7.4.2 Implementation details

For experiments on Long-Tailed CIFAR-10/100 [84] and CelebA-5 [240], we perform training with ResNet-32 [4] for 200 epochs, using SGD with a momentum of 0.9, and a weight decay of 0.0002. We set the initial learning rate as 0.1, then decay by 0.01 at the 160th epoch and again at the 180th epoch. For fair comparison, We also use linear warm-up learning rate schedule [241] for the first 5 epochs. For data augmentation in training, we use the commonly used version: 4 pixels are padded on each side, and a 32×32 crop is randomly sampled from the padded image or its horizontal flip. For experiments under the setting of Balanced Softmax [7], we use Nesterov SGD with momentum 0.9 and weight-decay 0.0005 for training. We use a total mini-batch size of 512 images on a single GPU. The learning rate increased from 0.05 to 0.1 in the first 800 iterations. Cosine scheduler [242] is applied afterward, with a minimum learning rate of 0. Our augmentation follows Balanced Softmax and Equalization Loss [243]. In testing, the image size is 32×32. In end-to-end training, the model is trained for 13K iterations. In decoupled training experiments, we fix the Softmax model, i.e., the instance-balanced baseline model obtained from the standard end-to-end training, as the feature extractor. And the classifier is trained for 2K iterations.

For Places-LT, we choose ResNet-152 as the backbone network and pretrain it on the full ImageNet-2012 dataset, following Decoupled training [91]. We use SGD optimizer with momentum 0.9, batch size 512, cosine learning rate schedule [242]. Similar to Decoupled training, we fine-tune the backbone with Instance-balanced sampling for representation learning and then re-train the classifier with our proposed algorithm with class-balanced sampling. Through the chapter, we refer to decoupled training as training the last linear classifier on a fixed feature extractor obtained from instance-balanced training. For experiments with the SSP method, we use rotation prediction [244] as self-supervised learning method, where an image is rotated by a random multiple of 90 degrees and a model is trained to predict the rotation degree as a 4-way classification problem.

We conduct all the experiments on NVIDIA GeForce RTX 3090, and implement all methods with default parameters by PyTorch [245]. All experiments are repeated five times with different seeds and we report the average test accuracy. We tune the hyperparameter η on the validation set, then train the model on the full training set. The α is fixed as $(\max_j \beta_j + \min_j \beta_j)$ by default and we find this value performs

TABLE 7.1: Test accuracy (%) of ResNet-32 on long-tailed CIFAR-10 and CIFAR-100 with various imbalance ratios. "†" indicates the reported results from [6]. The bold indicates the improved results by integrating our regularization.

Dataset	Long-tailed CIFAR-10			Long-tailed CIFAR-100		
Imbalance Ratio	100	50	10	100	50	10
Standard	71.61 ± 0.21	77.30 ± 0.13	86.74 ± 0.41	37.59 ± 0.19	43.20 ± 0.30	56.44 ± 0.12
SMOTE †	71.50 ± 0.57	-	85.70 ± 0.25	34.00 ± 0.33	-	53.80 ± 0.93
CB-RW	72.57 ± 1.30	78.19 ± 1.79	87.18 ± 0.95	38.11 ± 0.78	43.26 ± 0.87	56.40 ± 0.40
CB-Focal	70.91 ± 0.39	77.71 ± 0.57	86.89 ± 0.21	37.84 ± 0.80	42.96 ± 0.77	56.09 ± 0.15
Ours	**77.62 ± 0.28**	**81.76 ± 0.51**	**89.38 ± 0.46**	**40.26 ± 0.65**	**44.77 ± 0.25**	**58.09 ± 0.29**
LDAM-RW	74.21 ± 0.61	78.86 ± 0.65	86.44 ± 0.78	29.02 ± 0.34	36.41 ± 0.84	54.23 ± 0.72
+ Ours	**75.19 ± 0.34**	**79.76 ± 0.44**	**87.28 ± 0.61**	**35.85 ± 0.62**	**42.18 ± 0.82**	**55.48 ± 0.59**
LDAM-DRW	78.08 ± 0.38	81.88 ± 0.44	87.49 ± 0.18	42.84 ± 0.25	47.13 ± 0.28	57.18 ± 0.47
+ Ours	**79.82 ± 0.31**	**82.22 ± 0.45**	**87.83 ± 0.38**	**44.07 ± 0.75**	**47.5 ± 0.24**	**57.43 ± 0.31**
Balanced Softmax	78.03 ± 0.28	81.63 ± 0.39	88.10 ± 0.32	42.11 ± 0.70	46.79 ± 0.24	58.06 ± 0.40
+ Ours	**79.05 ± 0.20**	**82.76 ± 0.52**	**88.89 ± 0.21**	**42.86 ± 0.27**	**47.28 ± 0.58**	**58.80 ± 0.72**
SSP	74.58 ± 0.16	79.20 ± 0.43	88.50 ± 0.24	43.00 ± 0.51	47.04 ± 0.60	59.08 ± 0.46
+ Ours	**79.38 ± 0.65**	**82.18 ± 0.33**	**88.80 ± 0.43**	**43.57 ± 0.29**	**48.66 ± 0.57**	**59.78 ± 0.91**

well overall. For the η in the training objective, the best value depends on the dataset, imbalance ratio, network architecture, and the integrated method.

For OOD detection task, we measure the following metrics: (1) the false positive rate (FPR95) of OOD instances when the true positive rate of in-distribution examples is at 95%; (2) the area under the receiver operating characteristic curve (AUROC); and (3) the area under the precision-recall curve (AUPR).

7.4.3 Comparison methods

In this section, we verify that Open-sampling can boost the standard training and several state-of-the-art techniques by integrating Open-sampling with the following methods: 1) Standard: all the examples have the same weights; by default, we use standard cross-entropy loss. 2) SMOTE [98]: a variant of re-sampling with data augmentation. 3) CB-RW [102]: training examples are re-weighted according to the inverse of the effective number of samples in each class, defined as $(1-\beta^{n_i})/(1-\beta)$. 4) CB-Focal [102]: the CB method is combined with Focal loss. 5) M2m [6]: an over-sampling method with adversarial examples. 6) LDAM-RW [110]: the method derives a generalization error bound for the imbalanced training and uses a margin-aware multi-class weighted cross entropy loss. 7) LDAM-DRW [110]: the network is trained with LDAM loss and deferred re-balancing training. 8) Balanced Softmax

TABLE 7.2: Classification accuracy (%) on CIFAR-10 under the setting of Balance Softmax [7]. The bold indicates the improved results by integrating our regularization. Baseline results are taken from Balance Softmax[7]. " ⋆" indicates decoupled training.

Imbalance Factor	200	100	10
Standard	71.2	77.4	90.0
CB-RW	72.5	78.6	90.1
LDAM-RW	73.6	78.9	90.3
Equalization Loss [243]	74.6	78.5	90.2
Balanced Softmax	79.0	83.1	90.9
Ours	80.6	83.6	90.6
cRT⋆ [91]	76.6	82.0	91.0
LWS⋆ [91]	78.1	83.7	**91.1**
Ours⋆	**81.1**	**84.9**	90.9

[7]: the method derives a Balanced Softmax function from the probabilistic perspective that explicitly models the test-time label distribution shift. 9) SSP [112]: the method uses self-supervised learning to pre-train the network on the auxiliary dataset before standard training. Here, we do not expect vanilla Open-sampling to achieve state-of-the-art results compared with many complicated methods, our method can be still a promising option in the family of class-imbalanced learning methods, because it can outperform existing data re-balancing methods and improve existing state-of-the-art methods.

7.4.4 Main results

Results on long-tailed CIFAR. Extensive experiments are conducted on long-tailed CIFAR datasets with three different imbalance ratios: 10, 50, and 100. We use 300K Random Images[3] [3] as the open-set auxiliary dataset. The test accuracy of ResNet-32 [4] on long-tailed CIFAR datasets are reported in Table 7.1. The results show that our method can achieve impressive improvements on the standard training method. Especially for long-tailed CIFAR-10 with imbalance ratio 100, an extreme imbalance case, the vanilla Open-sampling can significantly outperform the standard baseline by 8.39%. Besides, incorporating our method into existing state-of-the-art methods can consistently improve their performance under various imbalance ratios.

[3]The dataset is published on `https://github.com/hendrycks/outlier-exposure`.

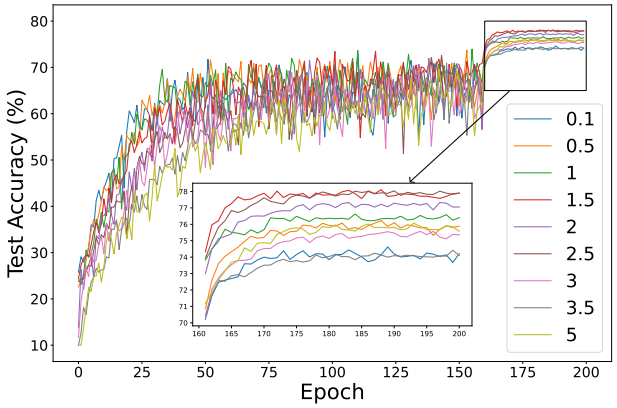


FIGURE 7.3: Results of sensitivity analysis on long-tailed CIFAR-10 with various values for η.

We also conduct experiments on long-tailed CIFAR-10 following the training setting of Balanced Softmax [7] and present the results in Table 7.2. Under this setting, our proposed method achieves the best performance among the end-to-end methods and could also improve the performance of decoupled training [91].

To further clarify the influence of η, we present a sensitivity analysis on long-tailed CIFAR-10 dataset with imbalance ratio 100 in Figure 7.3. We highlight the differences in the trend of test accuracy after the decay of learning rate at the 160th epoch. From the figure, we can observe that with a proper value of η like 1.5, the generalization performance can be largely improved by our proposed regularization. The result verifies that our regularization is an effective method to improve the generalization performance in long-tailed imbalanced learning.

Results on CelebA-5. We further verify the effectiveness of our method on real-world class-imbalanced datasets. The CelebA dataset has a long-tailed label distribution and the test set has a balanced label distribution. We use 300K Random Images [3] as the open-set auxiliary dataset. Table 7.3 summarizes test accuracy on the CelebA-5 dataset. In particular, the proposed method outperforms existing data-rebalancing methods and is able to consistently improve the existing state-of-the-art methods in test accuracy.

Results on Places-LT. For Places-LT, we follow the protocol of Decoupled training [91] and start from a ResNet-152 [4] backbone pre-trained on the ImageNet dataset [85]. We fine-tune the backbone with Instance-balanced sampling for representation learning and then re-train the classifier with our proposed algorithm as decoupled training. Here, we use Places-extra69 [86] as the open-set auxiliary

TABLE 7.3: Classification accuracy (%) on CelebA-5 with ResNet-32. "†" indicates the reported results are obtained from [6]. The shadow indicates the improved results.

Method	Accuracy	Method	Accuracy	Method	Accuracy
Standard	72.7	M2m †	75.6	LDAM-DRW	74.5
SMOTE †	72.8	**Ours**	**76.8**	**LDAM-DRW + Ours**	**76.9**
CB-RW	73.6	LDAM-RW	73.1	Balanced Softmax	76.4
CB-Focal	74.2	**LDAM + Ours**	**75.8**	**Balanced Softmax + Ours**	**78.6**

TABLE 7.4: Top-1 accuracy (%) on Places-LT with an ImageNet pre-trained ResNet-152. Baseline results are taken from original papers. "DT" indicates decoupled training.

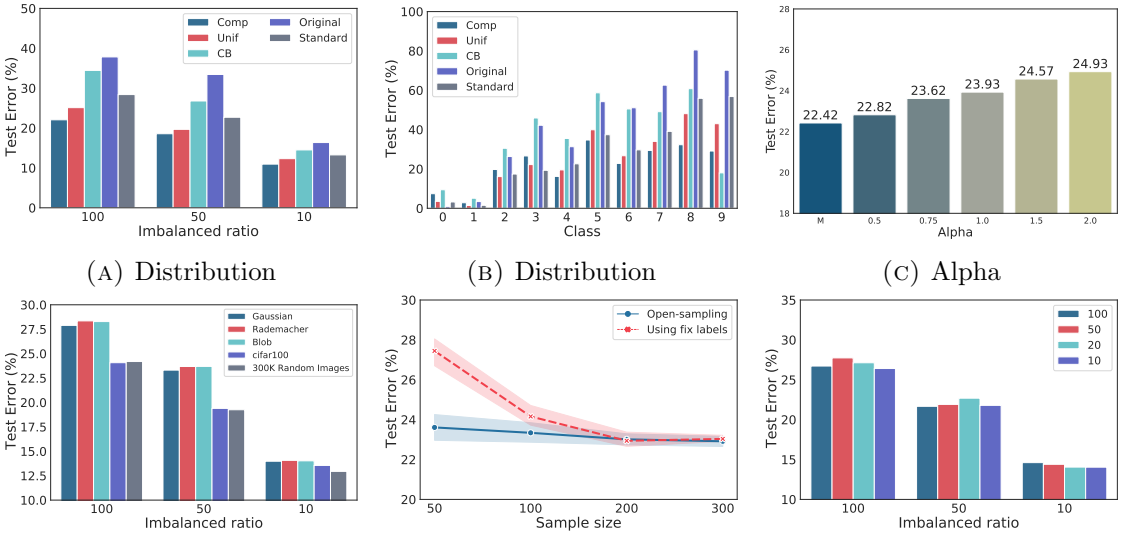
Method	Many	Medium	Few	Overall
Lifted Loss [246]	41.1	35.4	24.0	35.2
Focal Loss	41.1	34.8	22.4	34.6
Range Loss [247]	41.1	35.4	23.2	35.1
OLTR [109]	44.7	37.0	25.3	35.9
cRT⋆ [91]	42.1	37.6	24.9	36.7
LWS⋆ [91]	40.6	39.1	28.6	37.6
Ours⋆	42.9	**39.3**	26.8	**38.2**

TABLE 7.5: OOD detection performance comparison on long-tailed CIFAR-10. All values are percentages and are averaged over the ten test datasets. "↑" indicates larger values are better, and "↓" indicates smaller values are better. Bold numbers are superior results.

Method	Test Accuracy ↑	FPR95 ↓	AUROC ↑	AUPR ↑
MSP	71.83	56.1	75.2	32.71
OE	66.74	32.38	84.15	36.86
Ours	**77.62**	**20.68**	92.40	58.38
Ours ($\alpha = 5$)	75.16	22.13	**94.26**	**75.91**

dataset. Table 7.4 summarizes test accuracy on the Places-LT dataset. The proposed method achieves the best overall performance, show that our method and decoupled training scheme are two orthogonal components and Open-sampling is applicable for large-scale datasets.

OOD detection under class imbalance. Out-of-distribution (OOD) detection is an essential problem for the deployment of deep learning especially in safety-critical applications [3, 25, 31, 248] and it would be more difficult when the training dataset exhibits long-tailed distributions. Outlier Exposure (OE) [3] is a commonly-used method to improve anomaly detection performance with auxiliary dataset. Specifically, the training objective of OE under class imbalance is



(A) Distribution (B) Distribution (C) Alpha

(D) Auxiliary data (E) Sample size (F) Class number

FIGURE 7.4: Analytical experiments of Open-sampling on long-tailed CIFAR-10: (7.4a)(7.4b) with various label distributions, (7.4c) with various values of the α, (7.4d) with various open-set auxiliary datasets, including simulated noise datasets and real-world datasets. All the datasets contain 50,000 instances. (7.4e) with various sample sizes (K) of the auxiliary dataset, (7.4f) with various number of classes in the auxiliary datasets that are randomly sampled from CIFAR-100. Experiments in (7.4b), (7.4c), and (7.4e) are conducted under the imbalance ratio 100. The y-axis represents the test error in all the figures. "Standard" denotes the baseline with standard cross-entropy loss.

$\mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{train}}}[-\log f_y(x)] + \lambda\mathbb{E}_{x\sim\mathcal{D}_{\text{out}}}[H(P(Y); f(x))]$, where H is the cross entropy and $P(Y)$ is the label prior of the original training dataset. Following OE [3], we consider the maximum softmax probability (MSP) baseline [115]. Here, we conduct experiments on long-tailed CIFAR10 with imbalance rate 100 to verify the advantage of Open-sampling on OOD detection under class imbalance. Table 7.5 presents the test accuracy and the average performance over the three types of noises and seven OOD test datasets. We can observe that our method achieves impressive improvement on both the test accuracy and the detection performance.

7.5 Discussion

To provide a comprehensive understanding of the proposed method, we conduct a set of analyses in this section. Firstly, we compare our defined distribution with several alternative distributions to show the advantage of the Complementary Distribution in our method. Secondly, the effect of α in our method is thoroughly

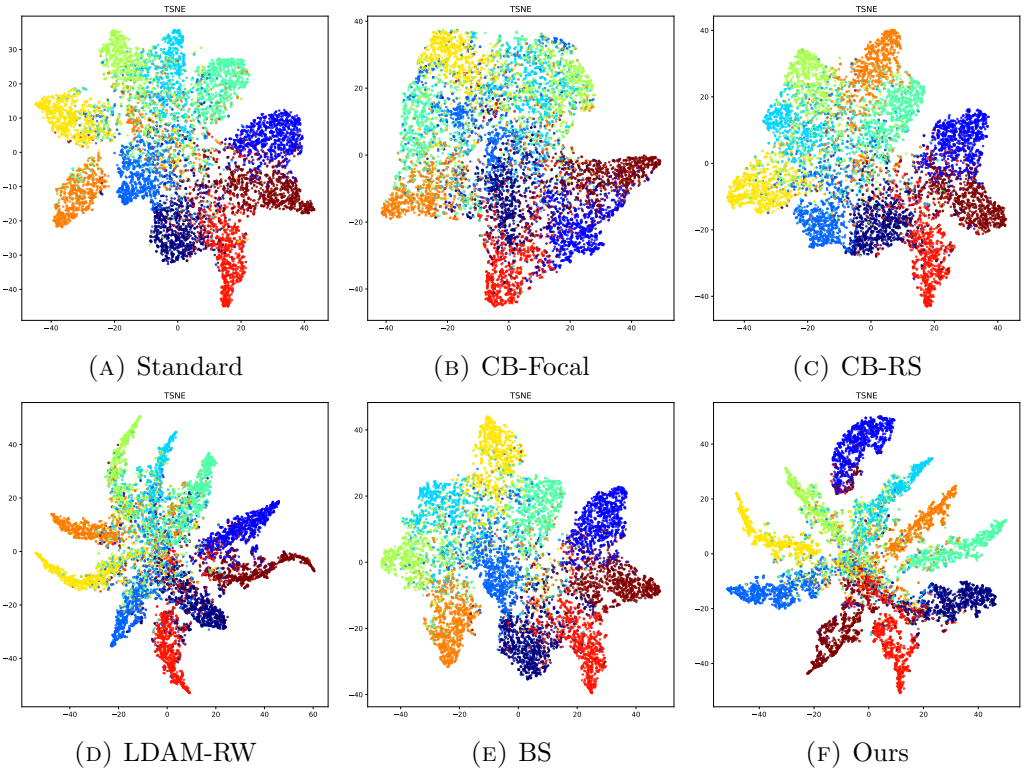
analyzed by empirical studies. Then, we present a guideline about how to choose or collect a suitable open-set auxiliary dataset for long-tailed imbalanced learning. Finally, we analyze the effect of the proposed method through the lens of decision boundaries.

The advantage of Complementary Distribution. In the proposed method, the labels of OOD instances from the auxiliary dataset are sampled from a random label distribution. For the random label distribution, we defined a Complementary Distribution in Proposition 7.1 and also presented several commonly used distributions, including uniform distribution (Unif), class balanced distribution (CB) [102], and the original class priors of the training dataset (Original). Here, we conduct experiments to compare the performance of the Open-sampling variants with different label distributions. The results in Figure 7.4a show that using the Complementary Distribution consistently achieves the best performance on the test set. In particular, using the uniform distribution can also improve the generalization performance while both CB and the original class priors deteriorate the performance of the neural networks.

To further understand why CB is not a good choice in our method, we present the per-class top-1 error on long-tailed CIFAR-10 in Figure 7.4b. Although using the CB distribution can achieve better performance on the smallest class, it downgrades the generalization performance on the other classes. The reason is that the CB distribution is far away from the uniform distribution, thereby introducing too much noise to the Bayes classifier. Different from CB, the Complementary Distribution is closer to a uniform distribution, making it achievable to re-balance the class priors while almost keeping non-toxicity of the open-set noisy labels.

Here, we also show the effect of α in Figure 7.4c. As analyzed in Section 7.2, the larger the value of α is, the Complementary Distribution tends to be closer to a uniform distribution. Here, "M" denotes the default value of $\alpha = (\max_j \beta_j + \min_j \beta_j)$. Additionally, the MCD variant achieves 22.51% on the test error. From Figure 7.4c, the test error presented a slightly upward trend with the increasing of the value of α. The results verified that it is necessary to adopt a complementary distribution, instead of simply using a uniform distribution.

The choices of open-set auxiliary dataset. With proper label distribution, can any OOD dataset be used to improve Long-tail imbalanced learning? In Figure



(A) Standard (B) CB-Focal (C) CB-RS

(D) LDAM-RW (E) BS (F) Ours

FIGURE 7.5: t-SNE visualization of test set on long-tailed CIFAR-10 with imbalance ratio 100. We can observe that LDAM and our method appear to learn more separable representations than Standard training and the other algorithms.

7.4d, we show the test error on long-tailed CIFAR-10 with imbalance ratio 100 using Open-sampling with different auxiliary datasets. We can observe that using simple noises like Gaussian noise could not achieve similar performance than those of real-world datasets, *e.g.*, CIFAR100 and 300K Random Images [3].

The sample size is another important factor for the open-set auxiliary dataset. In Figure 7.4e, we show that the performance of Open-sampling would be slightly better with a larger sample size of the auxiliary dataset. It is worth noting that the phenomenon is not consistent with that of [238], where using larger open-set auxiliary dataset could not improve the performance on learning with noisy labels. Here, we also conduct experiments using a variant of Open-sampling with fixed labels for OOD instances. We can observe that the variant performs poorly with a small auxiliary dataset, but the performance can be significantly improved by increasing the sample size of auxiliary dataset. In particular, the variant could achieve nearly the same improvement as the proposed Open-sampling method, with a large enough sample size.

In addition, we also find that the diversity of the open-set auxiliary dataset is unimportant for the effectiveness of Open-sampling. We conduct experiments on long-tailed CIFAR-10 with imbalance ratio 100 and use the subset of CIFAR100 with different number of classes as the open-set auxiliary dataset. The sample size of these subsets are fixed as 5,000. The results of using different subsets are almost the same, achieving 73.28% test accuracy. The results are consistent with that in Figure 7.4d, where using CIFAR100 as open-set auxiliary dataset can achieve almost the same improvements as 300K Random Images [3].

The effect of Open-sampling as regularization. In addition to re-balancing the class prior, what is the effect of Open-sampling as a regularization method? To gain additional insight, we look at the t-SNE projection of the learnt representations for different algorithms in Figure 7.5. For each method, the projection is performed over test data. The figures show that the decision boundaries of Open-sampling and LDAM [110] are much clearer than those of the other methods. The phenomenon illustrates that Open-sampling can encourage the minority classes to have a larger margin, which is similar to the effect of the LDAM loss [110]. As shown in Section 7.4, our method could still boost the performance of the LDAM method, which demonstrates the differences between our works.

7.6 Related Work

Re-sampling. Re-sampling methods aims to re-balance the class priors of the training dataset. Under-sampling methods remove examples from the majority classes, which is infeasible under extremely data imbalanced settings [93, 94]. The over-sampling method adds repeated samples for the minority classes, usually causing over-fitting to the minority classes [95–97]. Some methods utilize synthesized in-distribution samples to alleviate the over-fitting issue but introduce extra noise [6, 98, 99]. In contrast to in-distribution samples used in existing methods, our approach exploits OOD instances to re-balance the class priors of the training dataset.

Re-weighting. Re-weighting methods propose to assign adaptive weights for different classes or samples. Generally, the vanilla scheme re-weights classes proportionally to the inverse of their frequency [100]. Focal loss [101] assigns low weights

to the well-classified examples. Class-balanced loss [102] proposes to re-weight by the inverse effective number of samples. However, these re-weighting methods tend to make the optimization of DNNs difficult under extremely data imbalanced settings [103, 104].

Other methods for long-tailed datasets. In addition to the data re-balancing approaches, some other solutions are also applied for class-imbalanced learning, including transfer-learning based methods [105, 106], two-stage training methods[91, 107–109], training objective based methods [7, 110, 111], expert methods [20] and self-supervised learning methods [112, 113]. For example, transfer-learning based methods addressed the class-imbalanced issue by transferring features learned from head classes with abundant training instances to under-represented tail classes [105, 106]. Two-stage training methods proposed to apply decoupled training where the classifier is re-balanced during the fine-tuning stage [91, 107–109]. Generally, our proposed method is complementary to these existing methods and our method can further improve their performance, which are explicitly shown in our experiments.

Utilizing auxiliary dataset. In the deep learning community, auxiliary dataset is utilized in various contexts, e.g., adversarial machine learning [249, 250] and weakly supervised learning [59, 251–258]. For example, OE [3] regularized the network to give conservative predictions on OOD instances. OAT [250] utilized OOD instances to improve generalization in adversarial training. ODNL [238] proposed to use open-set auxiliary dataset to prevent the model from over-fitting inherent noisy labels. In long-tailed imbalanced learning, recent work [112] introduced unlabeled-in-distribution data to compensate for the lack of training samples. Another work [259] adopted semi-supervised learning techniques to improve performance on long-tailed datasets with out-of-class images from related classes, which are different from out-of-distribution data in our method. To the best of our knowledge, we are the first to explore the benefits of *out-of-distribution data* in learning with long-tailed datasets.

OOD detection. OOD detection is an essential building block for safely deploying machine learning models in the open world [3, 31, 115, 193]. A common baseline for OOD detection is the softmax confidence score [115]. It has been theoretically shown that neural networks with ReLU activation can produce arbitrarily high softmax confidence for OOD inputs [260]. To improve the performance, previous

methods have explored using artificially synthesized data from GANs [261] or unlabeled data [3, 126] as auxiliary OOD training data. Energy scores are shown to be better for distinguishing in- and out-of-distribution samples than softmax scores [31]. As a side effect, our method could also achieve superior performance in OOD detection tasks even if the labels of training dataset are noisy.

7.7 Chapter Summary

In this chapter, we propose a simple yet effective method termed Open-sampling, by introducing OOD instances to re-balance the class priors of the training dataset. To the best of our knowledge, our method is the first to utilize OOD instances in long-tailed learning. We show that our method not only re-balances the training dataset, but also promotes the neural network to learn more separable representations. Besides, we also present a guideline about the open-set auxiliary datasets: the realism and sample size are more important than the diversity. Extensive experiments demonstrate that our proposed method can enhance the performance of existing state-of-the-art methods, and also achieves strong OOD detection performance under class-imbalanced setting. Overall, our method enables to exploit OOD data for long-tail learning and we expect that our insights inspire future research to further explore the benefits of auxiliary datasets.

Chapter 8

Conclusion and Future Directions

8.1 Conclusions

Robust deep learning is an increasingly important topic for deploying machine learning models in the real world, and has attracted a surge of interest. In this thesis, we studied three advanced topics in robust deep learning, i.e., label noise, distribution shift, and exploring the benefits of OOD instances in training robustness. In the following, we present a conclusion for each robust algorithm in the previous chapters.

In Chapter 3, we propose an effective approach called JoCoR to improve the label-noise robustness with the agreement maximization principle, which reduces the diversity of two networks during networks. It breaks the conventional wisdom that keeping disagreement among networks is critical for robust learning with noisy labels. Our method utilizes a joint loss with Co-Regularization for both the parameter updating and example selection. Empirical results verified that JoCoR is superior to many state-of-the-art approaches. In future work, we may explore the theoretical foundation of agreement maximization principle based on the view of traditional Co-training algorithms [152, 153].

In Chapter 4, we propose a novel end-to-end deep learning method named UnionNet, which is not only theoretically consistent but also experimentally effective and efficient. We prove the learning consistency of UnionNet by establishing an estimation error bound, which shows that obtained empirical risk minimizer by UnionNet

would converge to the true risk minimizer as the amount of training data tends to infinity. Experimental results demonstrate the effectiveness of UnionNet in model generalization and training efficiency.

In Chapter 5, we introduce LogitNorm – a simple and effective alternative to the cross-entropy loss, which decouples the influence of logits' norm from the training procedure. Empirical results show that LogitNorm can improve both OOD detection and confidence calibration while maintaining the classification accuracy on in-distribution data. It is straightforward to implement with existing deep learning frameworks, and does not require sophisticated changes to the loss or training scheme. We hope that our insights inspire future research to further explore loss function design for OOD detection.

In Chapter 6, we provide the first attempt to improve learning with noisy labels with out-of-distribution data. In particular, we proposed Open-set regularization with Dynamic Noisy Labels (ODNL), a novel technique that enhances many existing robust training methods for label-noise robustness. We show that ODNL can be considered as a variant of SGD noises that usually leads to a flat minimum. Extensive experiments show that ODNL can help improve the robustness against various types of noisy labels and improve the OOD detection performance.

In Chapter 7, we propose a simple yet effective method termed Open-sampling, by introducing OOD instances to re-balance the class priors of the training dataset. Open-sampling is the first to utilize OOD instances in long-tailed learning. Open-sampling not only re-balances the training dataset, but also promotes the neural network to learn more separable representations. Besides, we also present a guideline about the open-set auxiliary datasets. Experiments validate the effectiveness of Open-sampling in both model generalization and OOD detection.

8.2 Future Directions

In this thesis, we have provided a basic blueprint of natural robustness with several novel algorithms across the areas of weakly supervised learning and open-world machine learning. Here, we present some potentially interesting future directions of robust deep learning.

8.2.1 Robustness problems in foundation models

Foundation models (like GPT-3 [262], BERT [263], and CLIP [264]) have become an increasingly important component in the deep learning community. Instead of training from scratch, the foundation model is trained on broad data (not task-specific) that can be adapted to a wide range of downstream tasks. While these models have been widely adopted in various applications, their potential robustness issues have not been explored, which may make them especially vulnerable to the natural occurring corruptions. A potential challenge for robustness in foundation models is from the demanding of huge computational resource, which makes it challenging to retrain or update the model for stronger robustness. A possible solution for this challenge is to explore methods in the form of model patching so that we can update the model with minimum training cost.

8.2.2 Moving from static to dynamic environments

Most existing research in natural robustness so far focuses on solving issues in static environments, where the models are fixed after deployment. In the real-world cases, the data streaming is continually evolving and unpredictable, which makes it difficult to maintain the model reliability. Therefore, it is of great significance to form a virtuous circle in the learning system by adapting to new challenges from the dynamic environment.

8.2.3 Theoretical understanding of OOD data

In Chapter 6 and Chapter 7, we have demonstrated the value of OOD data in remedying the imperfect training data and provided theoretical explanations from the perspectives of SGD noise and Bayesian inference. However, there still arises many questions: When OOD data can help? How much OOD data can help? Are there any rigorous conditions for OOD data to be benign in the training? .... To answer these questions, we need to construct theoretical frameworks for OOD data in the training dynamics.

Appendix A

Proofs for Chapter 4

A.1 Proof of Lemma 4.1

If the cross entropy loss is used, we have the following optimization problem:

$$\begin{aligned} \phi(g) = -\sum\nolimits_{i=1}^{k} p(y=i \mid \boldsymbol{x}) \log(g_i(\boldsymbol{x})) \\ \text{s.t. } \sum\nolimits_{i=1}^{k} g_i(\boldsymbol{x}) = 1. \end{aligned}$$

By using the Lagrange multiplier method, we can obtain the following non-constrained optimization problem:

$$\begin{aligned} \Phi(g) = -\sum\nolimits_{i=1}^{k} p(y=i \mid \boldsymbol{x}) \log(g_i(\boldsymbol{x})) \\ + \lambda(\sum\nolimits_{i=1}^{k} g_i(\boldsymbol{x}) - 1)). \end{aligned}$$

By setting the derivative to 0, we obtain

$$g_i^\star(\boldsymbol{x}) = \frac{1}{\lambda} p(y=i \mid \boldsymbol{x}).$$

Because $\sum_{i=1}^{k} g_i^\star(\boldsymbol{x}) = 1$ and $\sum_{i=1}^{k} p(y=i \mid \boldsymbol{x}) = 1$, we have

$$\sum\nolimits_{i=1}^{k} g_i^\star(\boldsymbol{x}) = \frac{1}{\lambda} \sum\nolimits_{i=1}^{k} p(y=i \mid \boldsymbol{x}) = 1.$$

Therefore, we can easily obtain $\lambda = 1$. In this way, $g_i^\star = \frac{1}{\lambda} p(y=i \mid \boldsymbol{x}) = p(y=i \mid \boldsymbol{x})$, which concludes the proof of Lemma 4.1. □

A.2 Proof of Theorem 4.2

According to Lemma 1, when we obtain $\widetilde{f}^\star$ by optimizing Eq. (4), we could optimally fit the conditional probability $p(\widetilde{\boldsymbol{y}} \mid \boldsymbol{x})$:

$$q^\star(\boldsymbol{x}) = p(\widetilde{\boldsymbol{y}} \mid \boldsymbol{x}), \forall i \in [k].$$

Let us introduce $v(\boldsymbol{x}) = p(y \mid \boldsymbol{x})$ and $\overline{v}(\boldsymbol{x}) = p(\widetilde{y} \mid \boldsymbol{x})$. According to Eq. (3), we have

$$\overline{v}(\boldsymbol{x}) = \boldsymbol{T}v(\boldsymbol{x}).$$

Since $q^\star(\boldsymbol{x}) = \overline{v}(\boldsymbol{x})$, we have $q^\star(\boldsymbol{x}) = \boldsymbol{T}v(\boldsymbol{x})$, On the other hand, we obtain $\widetilde{g}^\star(\boldsymbol{x})$ ($\widetilde{g}^\star(\boldsymbol{x}) = \text{softmax}(\widetilde{f}^\star(\boldsymbol{x}))$) by optimizing Eq. (4). Thus $q^\star(\boldsymbol{x}) = \boldsymbol{T}\widetilde{g}^\star(\boldsymbol{x})$, which further ensures that $\boldsymbol{T}v(\boldsymbol{x}) = \boldsymbol{T}\widetilde{g}^\star(\boldsymbol{x})$. In this way, if the transition matrix $\boldsymbol{T}$ is invertible, we can obtain $v(\boldsymbol{x}) = \widetilde{g}^\star(\boldsymbol{x})$. As we have shown that $v(\boldsymbol{x}) = p(y \mid \boldsymbol{x}) = g^\star(\boldsymbol{x})$ where $g^\star(\boldsymbol{x}) = \text{softmax}(f^\star(\boldsymbol{x}))$, we can obtain $\widetilde{g}^\star(\boldsymbol{x}) = g^\star(\boldsymbol{x})$, which implies that $\widetilde{f}^\star = f^\star$. Thus, Theorem 4.2 is proved. □

A.3 Proof of Theorem 4.3

Before proving Theorem 2, we first introduce the following lemma.

Lemma A.1. *The following inequality holds:*

$$R(\widehat{f}) - R(f^\star) \leq 2 \sup_{f \in \mathcal{F}} \left| R(f) - \widehat{R}(f) \right|.$$

Proof.

$$\begin{aligned} R(\widehat{f}) - R(f^\star) &\leq R(\widehat{f}) - \widehat{R}(\widehat{f}) + \widehat{R}(\widehat{f}) - R(\widetilde{f}^\star) \\ &\leq R(\widehat{f}) - \widehat{R}(\widehat{f}) + R(\widehat{f}) - R(\widetilde{f}^\star) \\ &\leq 2 \sup_{f \in \mathcal{F}} \left| R(f) - \widehat{R}(f) \right|, \end{aligned}$$

where we used the notations $\widetilde{f}^\star = \arg\min_{f\in\mathcal{F}} R(f)$ and $\widehat{f} = \arg\min_{f\in\mathcal{F}} \widehat{R}(f)$, and $\widehat{R}(f)$ represents the objective in Eq. (6) of UnionNet-B, which is an empirical approximation of $R(f)$. □

Then, we introduce the *uniform deviation bound*, which is useful to derive estimation error bounds. The proof can be found in some textbooks such as [183].

Lemma A.2. *Let Z be a random variable drawn from a probability distribution with density μ, $\mathcal{H} = \{h : \mathcal{Z} \mapsto [0, M]\}$ ($M > 0$) be a class of measurable functions, $\{z_i\}_{i=1}^n$ be i.i.d. examples drawn from the distribution with density μ. Then, for any delta > 0, with probability at least $1-\delta$,*

$$\sup_{h\in\mathcal{H}} \left| \mathbb{E}_{Z\sim\mu}\big[h(Z)\big] - \frac{1}{n}\sum_{i=1}^{n} h(z_i) \right| \leq 2\mathfrak{R}_n(\mathcal{H}) + M\sqrt{\frac{\log\frac{2}{\delta}}{2n}}.$$

Based on Lemma A.2, we can directly obtain the following lemma.

Lemma A.3. *Assume $\widetilde{\mathcal{L}}$ is upper bounded by M, i.e., for any $(\boldsymbol{x}, \widetilde{\boldsymbol{y}}) \sim p(\boldsymbol{x}, \widetilde{\boldsymbol{y}})$, $\widetilde{\mathcal{L}}(f(\boldsymbol{x}), \widetilde{\boldsymbol{y}}) \leq M$. Then, for any $\delta > 0$, with probability at least $1-\delta$,*

$$R(\widehat{f}) - R(f^\star) \leq 2\mathfrak{R}_n(\widetilde{\mathcal{L}} \circ \mathcal{F}) + M\sqrt{\frac{\log\frac{2}{\delta}}{2n}},$$

where $\widetilde{\mathcal{L}} \circ \mathcal{F}$ is defined as $\widetilde{\mathcal{L}} \circ \mathcal{F} = \{\widetilde{\mathcal{L}} \circ f \mid f \in \mathcal{F}\}$.

Then, we need to upper bound $\mathfrak{R}_n(\widetilde{\mathcal{L}} \circ \mathcal{F})$. Since $\widetilde{\mathcal{L}}$ is assumed to be ρ-Lipschitz continuous w.r.t. $f(\boldsymbol{x})$, according to the *Rademacher vector contraction inequality* [265], we have

$$\widehat{\mathfrak{R}}_n(\widetilde{\mathcal{L}} \circ \mathcal{F}) \leq \sqrt{2}\rho \sum_{y=1}^{k} \widehat{\mathfrak{R}}_n(\mathcal{F}_y),$$

where $\widehat{\mathfrak{R}}_n(\mathcal{F})$ and $\widehat{\mathfrak{R}}_n(\mathcal{F}_y)$ are the *empirical Rademacher complxity* [183] of $\mathcal{F}$ and $\mathcal{F}_y$. By taking the expectation over $p(\boldsymbol{x})$, we have $\mathfrak{R}_n(\widetilde{\mathcal{L}} \circ \mathcal{F}) \leq \sqrt{2}\rho \sum_{y=1}^{k} \mathfrak{R}_n(\mathcal{F}_y)$. By further taking into account Lemma A.3, Theorem 4.3 is proved. □

Appendix B

Proofs for Chapter 5

B.1 Proof of Proposition 5.1

From Eq. (5.2), we have $\boldsymbol{f} = \|\boldsymbol{f}\| \cdot \hat{\boldsymbol{f}}$. Then,

$$\arg\max_i(f_i) = \arg\max_i(\|\boldsymbol{f}\| \cdot \hat{f}_i) = \arg\max_i(\hat{f}_i).$$

Similarly, for any given scalar $s > 1$, we have,

$$\arg\max_i(sf_i) = \arg\max_i(s\|\boldsymbol{f}\| \cdot \hat{f}_i) = \arg\max_i(\hat{f}_i).$$

Thus Proposition 5.1 is proved. □

B.2 Proof of Proposition 5.2

From Proposition 5.1, we have

$$\arg\max_i(f_i) = \arg\max_i(sf_i) = c.$$

Let $f_c = \max_i(f_i)$, and $t = s - 1$, then we have,

$$\begin{aligned}\sigma_c(s\boldsymbol{f}) &= \frac{e^{(1+t)f_c}}{\sum_{j=1}^{n} e^{(1+t)f_j}} \\ &= \frac{e^{f_c}}{\sum_{j=1}^{n} e^{f_j + t(f_j - f_c)}}.\end{aligned}$$

For any $j \in [1, 2, \ldots, n]$, we have, $f_j - f_c \leq 0$. Then

$$\sigma_c(s\boldsymbol{f}) \geq \frac{e^{f_c}}{\sum_{j=1}^{n} e^{f_j}} = \sigma_c(\boldsymbol{f}).$$

Thus Proposition 5.2 is proved. □

B.3 Proof of Proposition 5.3

Let $\widetilde{\boldsymbol{f}} = \boldsymbol{f}/(\tau\|\boldsymbol{f}\|)$, then we have $\|\widetilde{\boldsymbol{f}}\| = 1/\tau$.

That is, $\sum_{i=1}^{k} \widetilde{\boldsymbol{f}}_i^{\,2} = \|\widetilde{\boldsymbol{f}}\|^2 = 1/\tau^2$.

Hence,

$$-1/\tau \leq \widetilde{\boldsymbol{f}}_i \leq 1/\tau, \forall\ i \in 1, \ldots, k.$$

Let $\sigma(\widetilde{\boldsymbol{f}}) = \frac{e^{\widetilde{f}_y}}{\sum_{i=1}^{k} e^{\widetilde{f}_i}}$, then we have

$$\begin{aligned}\sigma(\widetilde{\boldsymbol{f}}) &\leq \frac{e^{1/\tau}}{e^{1/\tau} + (k-1)e^{-1/\tau}} \\ &= \frac{1}{1 + (k-1)e^{-2/\tau}}\end{aligned}$$

Hence,

$$\begin{aligned}\mathcal{L}_{\text{logit\_norm}} &= -\log(\sigma(\widetilde{\boldsymbol{f}})) \\ &\geq -\log\frac{1}{1 + (k-1)e^{-2/\tau}} \\ &= \log(1 + (k-1)e^{-2/\tau})\end{aligned}$$

Thus Proposition 5.3 is proved. □

Appendix C

Proofs for Chapter 6

C.1 Proof of Proposition 6.1

For Eq. 6.6, the noisy gradient is

$$\widetilde{\nabla}_{\boldsymbol{\theta}}\ell_{\text{total}} = \nabla_{\boldsymbol{\theta}}\ell\left(\boldsymbol{f}, y\right) + \eta \cdot \nabla_{\boldsymbol{\theta}}\ell\left(\widetilde{\boldsymbol{f}}, \widetilde{y}\right) = \nabla_{\boldsymbol{\theta}}\ell\left(\boldsymbol{f}, y\right) + \eta \cdot \nabla_{\widetilde{\boldsymbol{f}}}\ell\left(\widetilde{\boldsymbol{f}}, \widetilde{y}\right) \cdot \nabla_{\boldsymbol{\theta}}\widetilde{\boldsymbol{f}}$$

For convenience, we omit the hyperparameter η. Note that cross-entropy loss is $\ell\left(\boldsymbol{f}, y\right) = -\boldsymbol{e}^{y} \log \boldsymbol{f}$, then the noise on $\nabla_{\boldsymbol{\theta}}\ell\left(\boldsymbol{f}, y\right)$ is

$$\boldsymbol{z} = \nabla_{\widetilde{\boldsymbol{f}}}\ell\left(\widetilde{\boldsymbol{f}}, \widetilde{y}\right) \cdot \nabla_{\boldsymbol{\theta}}\widetilde{\boldsymbol{f}} = -\left(\frac{e^{\widetilde{y}}}{\widetilde{\boldsymbol{f}}}\right)^{T} \cdot \nabla_{\boldsymbol{\theta}}\widetilde{\boldsymbol{f}} = -\sum_{j=1}^{k}\left(e_{j}^{\widetilde{y}} \cdot \frac{\nabla_{\boldsymbol{\theta}_i}\widetilde{\boldsymbol{f}}_j}{\widetilde{\boldsymbol{f}}_j}\right), \quad \text{s.t. } \widetilde{y} \sim \mathcal{U}_k.$$

Since $\boldsymbol{e}^{\widetilde{y}} = (0, \cdots, 1, \cdots, 0)$ is the one-hot vector and only the y-th entry is 1, the expression of the noise $\boldsymbol{z}$ can be simplified as:

$$\boldsymbol{z} = -\frac{\nabla_{\boldsymbol{\theta}}\widetilde{\boldsymbol{f}}_j}{\widetilde{\boldsymbol{f}}_j}, \quad \text{s.t. } j \sim \mathcal{U}_k.$$

Let z_i be the i-th entry of $\boldsymbol{z}$, we have

$$z_i = -\frac{\nabla_{\boldsymbol{\theta}_i}\widetilde{\boldsymbol{f}}_j}{\widetilde{\boldsymbol{f}}_j}, \quad \text{s.t. } j \sim \mathcal{U}_k.$$

Hence,

$$\mathbb{E}\left(z_i\right) = -\frac{1}{k}\sum_{j=1}^{k}\frac{\nabla_{\boldsymbol{\theta}_i}\widetilde{\boldsymbol{f}}_j}{\widetilde{\boldsymbol{f}}_j}$$

□

C.2 Proof of Proposition 6.3

The regularization item of OE is $\ell_{\mathrm{OE}} = -\frac{1}{k}\cdot\sum_{i=1}^{k}\log\widetilde{\boldsymbol{f}}_i$.

Then the gradient of ℓ_{OE} w.r.t θ is

$$\nabla_\theta \ell_{\mathrm{OE}} = \nabla_{\widetilde{\boldsymbol{f}}}\ell_{\mathrm{OE}}\cdot\nabla_\theta\widetilde{\boldsymbol{f}} = \nabla_{\widetilde{\boldsymbol{f}}}\left(-\frac{1}{k}\cdot\sum_{j=1}^{k}\log\widetilde{\boldsymbol{f}}_j\right)\cdot\nabla_\theta\widetilde{\boldsymbol{f}} = -\frac{1}{k}\cdot\sum_{j=1}^{k}\frac{\nabla_\theta\widetilde{\boldsymbol{f}}_j}{\widetilde{\boldsymbol{f}}_j}$$

□

Appendix D

Proofs for Chapter 7

D.1 Proof of Theorem 7.1

Since $\mathcal{D}_{\text{mix}} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{out}}$, the underlying data distribution of $\mathcal{D}_{\text{mix}}$ will be a linear combination of the training distribution $P_{\text{s}}(X, Y)$ and the OOD distribution $P_{\text{out}}(X, Y)$:

$$P_{\text{mix}}(\boldsymbol{x}, y) = \frac{N}{M+N} P_{\text{s}}(\boldsymbol{x}, y) + \frac{M}{M+N} P_{\text{out}}(\boldsymbol{x}, y), \tag{D.1}$$

where N is the size of $\mathcal{D}_{\text{train}}$ and M is the size of $\mathcal{D}_{\text{out}}$.

By the virtue of Bayes' theorem, we have

$$\begin{aligned} & P_{\text{mix}}(\boldsymbol{x}|y) P_{\text{mix}}(y) \\ =& \frac{N}{M+N} P_{\text{s}}(\boldsymbol{x}|y) P_{\text{s}}(y) + \frac{M}{M+N} P_{\text{out}}(\boldsymbol{x}|y) P_{\text{out}}(y) \\ =& \frac{N}{M+N} P_{\text{s}}(\boldsymbol{x}|y) P_{\text{s}}(y) + \frac{M}{M+N} P_{\text{out}}(\boldsymbol{x}) P_{\text{out}}(y) \\ =& \frac{N}{M+N} P_{\text{s}}(\boldsymbol{x}|y) P_{\text{s}}(\boldsymbol{x}, y) + \frac{1}{K} \cdot \frac{M}{M+N} P_{\text{out}}(\boldsymbol{x}), \end{aligned} \tag{D.2}$$

where the second equality follows the fact that $P_{\text{mix}}(\boldsymbol{x}|y) = P_{\text{mix}}(\boldsymbol{x})$ since the label y is independent of the instance x for the OOD data, and the third equality is simply the fact that $P_{\text{out}}(y) = 1/K$.

Then, by taking the maximum of both sides, we have

$$\begin{aligned}
&\arg\max_{y\in\mathcal{Y}} P_{\text{mix}}(\boldsymbol{x}|y)P_{\text{mix}}(y) \\
=&\arg\max_{y\in\mathcal{Y}} \left\{\frac{N}{M+N}P_{\text{s}}(\boldsymbol{x}|y)P_{\text{s}}(\boldsymbol{x},y)+\frac{M/K}{M+N}P_{\text{out}}(\boldsymbol{x})\right\} \\
=&\arg\max_{y\in\mathcal{Y}} \frac{N}{M+N}P_{\text{s}}(\boldsymbol{x}|y)P_{\text{s}}(\boldsymbol{x},y). \\
=&\arg\max_{y\in\mathcal{Y}} P_{\text{s}}(\boldsymbol{x}|y)P_{\text{s}}(\boldsymbol{x},y).
\end{aligned} \tag{D.3}$$

Thus Theorem 7.1 is proved. □

D.2 Proof of Proposition 7.1

By definition, $\sum_{i=1}^{K}\Gamma_i = 1$ naturally holds for any α.

If $\alpha = \max_j(\beta_j)$, then $\Gamma_j = (\max_j(\beta_j) - \beta_j)/(K\cdot\max_j(\beta_j) - 1)$. In particular, for $k = \arg\max_i(\beta_i)$, we have $\Gamma_k = 0$.

In the case of $\alpha \to \infty$, let us denote $f(\alpha) = \alpha - \beta_j$ and $g(\alpha) = K\cdot\alpha - 1$. Since $\lim_{\alpha\to\infty} f(\alpha) = \lim_{\alpha\to\infty} g(\alpha) = \infty$, $g'(\alpha) = K \neq 0$, and $\lim_{\alpha\to\infty} f'(\alpha)/g'(\alpha) = 1/K$ exists, using L'Hôpital's rule, we have:

$$\lim_{\alpha\to\infty}\Gamma_j = \lim_{\alpha\to\infty}\frac{f(\alpha)}{g(\alpha)} = \lim_{\alpha\to\infty}\frac{f'(\alpha)}{g'(\alpha)} = 1/K$$

Thus Proposition 7.1 is proved. □

List of Author's Awards, Patents, and Publications[1]

Journal Articles

- Lei Feng, **Hongxin Wei**, Qingyu Guo, Zhuoyi Lin, Bo An. Embedding-Augmented Generalized Matrix Factorization for Recommendation with Implicit Feedback. *IEEE Intelligent Systems*, pp.32-41.
- **Hongxin Wei**, Renchunzi Xie, Lei Feng, Bo Han, Bo An. Deep Learning from Multiple Noisy Annotators as A Union. *IEEE Transactions on Neural Networks and Learning Systems*, accepted.
- Lei Feng, Senlin Shu, Yuzhou Cao, Lue Tao, **Hongxin Wei**, Tao Xiang, Bo An, Gang Niu. Multiple-Instance Learning from Unlabeled Bags with Pairwise Similarity. *Transactions on Knowledge and Data Engineering*, accepted.
- Eng Aik Chan, Carolina Rendón-Barraza, Benquan Wang, Tanchao Pu, Jun-Yu Ou, **Hongxin Wei**, Giorgio Adamo, Bo An, Nikolay I. Zheludev. Counting and mapping of subwavelength nanoparticles from a single shot scattering pattern. *Nanophotonics*, accepted.

Conference Proceedings

- **Hongxin Wei**, Lei Feng, Xiangyu Chen, Bo An. Combating noisy labels by agreement: A joint training method with co-regularization, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.13726-13735, 2020.

[1]The superscript * indicates joint first authors

- Rundong Wang*, **Hongxin Wei***, Bo An, Zhouyan Feng, Jun Yao. Commission Fee is not Enough: A Hierarchical Reinforced Framework for Portfolio Management. *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pp.626-633, 2021.

- Ziqi Zhang, Yuexiang Li, **Hongxin Wei**, Kai Ma, Tao Xu, Yefeng Zheng. Alleviating Noisy-label Effects in Image Classification via Probability Transition Matrix. *Proceedings of the 32nd British Machine Vision Conference*, 2021.

- Lei Feng, Senlin Shu, Yuzhou Cao, Lue Tao, **Hongxin Wei**, Tao Xiang, Bo An, Gang Niu. Multiple-Instance Learning from Similar and Dissimilar Bags. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data*, pp.374-382, 2021.

- **Hongxin Wei**, Lue Tao, Renchunzi Xie, Bo An. Open-set Label Noise Can Improve Robustness Against Inherent Label Noise. *Proceedings of the 35th Annual Conference on Neural Information Processing Systems*, pp.7978-7992, 2021.

- Renchunzi Xie, **Hongxin Wei**, Lei Feng, Bo An. GearNet: Stepwise Dual Learning for Weakly Supervised Domain Adaptation. *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, pp.8717-8725, 2022.

- **Hongxin Wei**, Lue Tao, Renchunzi Xie, Lei Feng, Bo An. Open-Sampling: Exploring Out-of-Distribution Data for Re-balancing Long-tailed Datasets. *Proceedings of the 39th International Conference on Machine Learning*, pp.23615-23630, 2022.

- **Hongxin Wei**, Renchunzi Xie, Hao Cheng, Lei Feng, Bo An, Yixuan Li. Mitigating Neural Network Overconfidence with Logit Normalization. *Proceedings of the 39th International Conference on Machine Learning*, pp.23631-23644, 2022.

- Huiping Zhuang, Zhenyu Weng, **Hongxin Wei**, Renchunzi Xie, Toh Kar-Ann, Zhiping Lin. Analytic Class-Incremental Learning with Absolute Memorization and Privacy Protection. *Proceedings of the 36th Annual Conference on Neural Information Processing Systems*, accepted, 2022.

- Lue Tao, Lei Feng, **Hongxin Wei**, Jinfeng Yi, Shengjun Huang, Songcan Chen. Can Adversarial Training Be Manipulated By Non-Robust Features? *Proceedings of the 36th Annual Conference on Neural Information Processing Systems*, accepted, 2022.

- Senlin Shu, Shuo He, Haobo Wang, **Hongxin Wei**, Tao Xiang, Lei Feng. A Generalized Unbiased Risk Estimator for Learning with Augmented Classes. *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, accepted.

- **Hongxin Wei**, Huiping Zhuang, Renchunzi Xie, Lei Feng, Gang Niu, Bo An, Yixuan Li. Mitigating Memorization of Noisy Labels by Clipping the Model Prediction. *Proceedings of the 40th International Conference on Machine Learning*, accepted, 2023.

Bibliography

[1] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference*, pages 87.1–87.12, 2016. xvii, xviii, xxi, xxii, 58, 59, 60, 61, 62, 63, 64, 66, 74, 82, 83

[2] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008. xviii, 60

[3] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *Proceedings of the International Conference on Learning Representations*, 2019. xix, 16, 68, 74, 80, 85, 91, 96, 99, 100, 101, 102, 104, 105, 106, 107

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. xxii, 17, 63, 64, 71, 97, 99, 100

[5] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017. xxii, 63, 64

[6] Jaehyung Kim, Jongheon Jeong, and Jinwoo Shin. M2m: Imbalanced classification via major-to-minor translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13896–13905, 2020. xxii, 14, 88, 95, 96, 98, 101, 105

[7] Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, Dec 2020. xxii, 14, 94, 97, 99, 100, 106

[8] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014. 1, 7

[9] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 1, 7

[10] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018. 1, 11

[11] Daniel W Otter, Julian R Medina, and Jugal K Kalita. A survey of the usages of deep learning for natural language processing. *IEEE transactions on neural networks and learning systems*, 32(2):604–624, 2020. 1, 11

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 1, 2, 11

[13] Nathan Drenkow, Numair Sani, Ilya Shpitser, and Mathias Unberath. Robustness in deep learning for computer vision: Mind the gap? *arXiv preprint arXiv:2112.00639*, 2021. 1, 11

[14] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017. 2

[15] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 2

[16] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on Computer and Communications Security*, pages 1322–1333, 2015. 2

[17] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017. 2

[18] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 2

[19] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021. 2, 14

[20] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella Yu. Long-tailed recognition by routing diverse distribution-aware experts. In *International Conference on Learning Representations*, 2021. 2, 14, 106

[21] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2018. 2, 9

[22] Jitendra Parmar, Satyendra Singh Chouhan, Vaskar Raychoudhury, and Santosh S Rathore. Open-world machine learning: applications, challenges, and opportunities. *ACM Computing Surveys (CSUR)*, 2021. 2, 10

[23] Yan Yan, Rómer Rosales, Glenn Fung, Ramanathan Subramanian, and Jennifer Dy. Learning from multiple annotators with varying expertise. *Machine Learning*, 95(3):291–327, 2014. 2, 11, 12, 17, 35, 71

[24] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM*, 50(4): 506–519, 2003. 2, 7, 11, 17, 71

[25] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021. 2, 3, 8, 14, 101

[26] Rohan Sinha, Apoorva Sharma, Somrita Banerjee, Thomas Lew, Rachel Luo, Spencer M Richards, Yixiao Sun, Edward Schmerling, and Marco Pavone. A system-level view on out-of-distribution data in robotics. *arXiv preprint arXiv:2212.14020*, 2022. 2

[27] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems*, pages 8527–8537, 2018. 2, 7, 12, 17, 18, 19, 20, 21, 24, 30, 38, 71, 77, 83

[28] Eran Malach and Shai Shalev-Shwartz. Decoupling "when to update" from "how to update". In *Advances in Neural Information Processing Systems*, pages 960–970, 2017. 12, 17, 24, 25, 82

[29] Xiyu Yu, Tongliang Liu, Mingming Gong, and Dacheng Tao. Learning with biased complementary labels. In *Proceedings of the European Conference on Computer Vision*, pages 68–83, 2018. 2, 11, 17, 39

[30] Filipe Rodrigues and Francisco C Pereira. Deep learning from crowds. In *Thirty-Second AAAI Conference on Artificial Intelligence*, pages 1611–1618, 2018. 3, 12, 34, 35, 37, 38, 41, 43, 45, 46, 48

[31] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3, 16, 53, 54, 56, 64, 68, 85, 101, 106, 107

[32] Kimin Lee, Sukmin Yun, Kibok Lee, Honglak Lee, Bo Li, and Jinwoo Shin. Robust inference via generative classifiers for handling noisy labels. In *Proceedings of International Conference on Machine Learning*, pages 3763–3772. PMLR, 2019. 3, 71, 75

[33] Yisen Wang, Weiyang Liu, Xingjun Ma, James Bailey, Hongyuan Zha, Le Song, and Shu-Tao Xia. Iterative learning with open-set noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8688–8696, 2018. 71, 82

[34] Ragav Sachdeva, Filipe R Cordeiro, Vasileios Belagiannis, Ian Reid, and Gustavo Carneiro. Evidentialmix: Learning with combined open-set and closed-set noisy labels. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3607–3615, 2021. 72

[35] Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Jiankang Deng, Jiatong Li, and Yinian Mao. Extended t: Learning with mixed closed-set and open-set noisy labels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 3, 71, 75

[36] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, 2020. 9, 10

[37] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems*, pages 1919–1930, 2019. 9, 12

[38] Jingfeng Wu, Wenqing Hu, Haoyi Xiong, Jun Huan, Vladimir Braverman, and Zhanxing Zhu. On the noisy gradient descent that generalizes as sgd. In *International Conference on Machine Learning*, pages 10367–10376. PMLR, 2020. 9, 72

[39] Jessa Bekker and Jesse Davis. Learning from positive and unlabeled data: A survey. *Machine Learning*, 109(4):719–760, 2020. 9

[40] Lei Feng, Jiaqi Lv, Bo Han, Miao Xu, Gang Niu, Xin Geng, Bo An, and Masashi Sugiyama. Provably consistent partial-label learning. *Advances in Neural Information Processing Systems*, 33:10948–10960, 2020. 9

[41] Min-Ling Zhang, Fei Yu, and Cai-Zhi Tang. Disambiguation-free partial label learning. *IEEE Transactions on Knowledge and Data Engineering*, 29(10): 2155–2167, 2017. 9

[42] Takashi Ishida, Gang Niu, Aditya Menon, and Masashi Sugiyama. Complementary-label learning for arbitrary losses and models. In *International Conference on Machine Learning*, pages 2971–2980. PMLR, 2019. 9

[43] Prissadang Suta, Xi Lan, Biting Wu, Pornchai Mongkolnam, and Jonathan H Chan. An overview of machine learning in chatbots. *International Journal of Mechanical Engineering and Robotics Research*, 9(4):502–510, 2020. 10

[44] Claudine Badue, Rânik Guidolini, Raphael Vivacqua Carneiro, Pedro Azevedo, Vinicius B Cardoso, Avelino Forechi, Luan Jesus, Rodrigo Berriel, Thiago M Paixao, Filipe Mutz, et al. Self-driving cars: A survey. *Expert Systems with Applications*, 165:113816, 2021. 10

[45] Yan Yan, Zhongwen Xu, Ivor Tsang, Guodong Long, and Yi Yang. Robust semi-supervised learning through label aggregation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016. 10

[46] Kaidi Cao, Maria Brbic, and Jure Leskovec. Open-world semi-supervised learning. *arXiv preprint arXiv:2102.03526*, 2021. 10

[47] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016. 10

[48] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020. 10

[49] Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 233–242, 2017. 11, 17, 21

[50] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016. 11, 17, 18, 21, 71, 76

[51] Filipe Rodrigues, Mariana Lourenco, Bernardete Ribeiro, and Francisco C Pereira. Learning supervised topic models for classification and regression from crowds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2409–2422, 2017. 12

[52] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11(4), 2010. 12, 33, 35, 38, 41

[53] Zhijun Chen, Huimin Wang, Hailong Sun, Pengpeng Chen, Tao Han, Xudong Liu, and Jie Yang. Structured probabilistic end-to-end learning from crowds. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 1512–1518, 2020. 12, 34, 35, 37, 38, 39, 45, 46

[54] Min-Ling Zhang and Zhi-Hua Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007. 12, 35

[55] Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8): 1819–1837, 2013.

[56] Hao Yang, Joey Tianyi Zhou, Yu Zhang, Bin-Bin Gao, Jianxin Wu, and Jianfei Cai. Exploit bounding box annotations for multi-label object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[57] Hao Yang, Joey Tianyi Zhou, Jianfei Cai, and Yew Soon Ong. Miml-fcn+: Multi-instance multi-label learning via fully convolutional networks with privileged information. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1577–1585, 2017.

[58] Hao Yang, Joey Tianyi Zhou, and Jianfei Cai. Improving multi-label learning with missing labels by structured semantic correlations. In *European Conference on Computer Vision*, pages 835–851. Springer, 2016. 12, 35

[59] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13726–13735, 2020. 12, 18, 38, 71, 77, 83, 106

[60] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor W Tsang, and Masashi Sugiyama. How does disagreement benefit co-teaching? *arXiv preprint arXiv:1901.04215*, 2019. 12, 17, 24

[61] Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In *International Conference on Machine Learning*, pages 312–321. PMLR, 2019. 12

[62] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020. 12, 84

[63] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. *arXiv preprint arXiv:1712.05055*, 2017. 12, 17, 71

[64] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):447–461, 2015. 17, 26

[65] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. *arXiv preprint arXiv:1803.09050*, 2018. 12, 17, 20

[66] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. *Advances in Neural Information Processing Systems*, 31, 2018. 12

[67] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1944–1952, 2017. 12, 23, 24, 38, 39, 71, 77, 82

[68] Pengfei Chen, Junjie Ye, Guangyong Chen, Jingwei Zhao, and Pheng-Ann Heng. Beyond class-conditional assumption: A primary attempt to combat instance-dependent label noise. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11442–11450, 2021. 12, 81, 82

[69] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014. 23, 71

[70] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5552–5560, 2018. 24

[71] Songzhu Zheng, Pengxiang Wu, Aman Goswami, Mayank Goswami, Dimitris Metaxas, and Chao Chen. Error-bounded correction of noisy labels. In *International Conference on Machine Learning*, pages 11447–11457. PMLR, 2020. 12

[72] Yueming Lyu and Ivor W. Tsang. Curriculum loss: Robust learning and generalization against label corruption. In *International Conference on Learning Representations*, 2020. 12

[73] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *International Conference on Machine Learning*, pages 6543–6553. PMLR, 2020.

[74] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330, 2019. 77

[75] Yilun Xu, Peng Cao, Yuqing Kong, and Yizhou Wang. L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. In *Advances in Neural Information Processing Systems*, pages 6222–6233, 2019.

[76] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in Neural Information Processing Systems*, pages 8778–8788, 2018. 12, 77, 81, 82

[77] Wei Hu, Zhiyuan Li, and Dingli Yu. Simple and effective regularization methods for training on noisily labeled data with generalization guarantee. *arXiv preprint arXiv:1905.11368*, 2019. 12

[78] Aditya Krishna Menon, Ankit Singh Rawat, Sanjiv Kumar, and Sashank Reddi. Can gradient clipping mitigate label noise? In *International Conference on Learning Representations (ICLR)*, 2020. 12, 82

[79] Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? In *Proceedings of International Conference on Machine Learning*, pages 6448–6458. PMLR, 2020. 12, 79, 80

[80] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016. 12, 69

[81] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations*, 2017. 12

[82] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993, 2018. 12

[83] Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. Self: Learning to filter noisy labels with self-ensembling. In *International Conference on Learning Representations*, 2020. 12

[84] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Tech Report*, 2009. 13, 61, 74, 87, 88, 95, 97

[85] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 13, 84, 87, 100

[86] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2017. 13, 62, 85, 87, 88, 95, 96, 100

[87] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 8769–8778, 2018.

[88] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014. 13, 87

[89] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9719–9728, 2020. 13, 87

[90] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2537–2546, 2019.

[91] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *Eighth International Conference on Learning Representations (ICLR)*, 2020. 13, 14, 87, 94, 97, 99, 100, 101, 106

[92] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 13

[93] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009. 13, 87, 105

[94] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449, 2002. 13, 87, 105

[95] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018. 14, 90, 105

[96] Jonathon Byrd and Zachary Lipton. What is the effect of importance weighting in deep learning? In *International Conference on Machine Learning*, pages 872–881. PMLR, 2019. 87

[97] Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *European Conference on Computer Vision*, pages 467–482. Springer, 2016. 14, 87, 105

[98] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002. 14, 87, 90, 98, 105

[99] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks*, pages 1322–1328, 2008. 14, 90, 105

[100] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5375–5384, 2016. 14, 105

[101] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017. 14, 105

[102] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 9268–9277, 2019. 14, 87, 98, 103, 106

[103] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 7032–7042, 2017. 14, 106

[104] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3111–3119, 2013. 14, 106

[105] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. Feature transfer learning for face recognition with under-represented data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5704–5713, 2019. 14, 106

[106] Muhammad Abdullah Jamal, Matthew Brown, Ming-Hsuan Yang, Liqiang Wang, and Boqing Gong. Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7610–7619, 2020. 14, 106

[107] Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 14, 106

[108] Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2361–2370, 2021.

[109] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 14, 101, 106

[110] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 14, 94, 98, 105, 106

[111] Youngkyu Hong, Seungju Han, Kwanghee Choi, Seokjun Seo, Beomsu Kim, and Buru Chang. Disentangling label distribution for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6626–6636, 2021. 14, 106

[112] Yuzhe Yang and Zhi Xu. Rethinking the value of labels for improving class-imbalanced learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 14, 87, 94, 99, 106

[113] Jiequan Cui, Zhisheng Zhong, Shu Liu, Bei Yu, and Jiaya Jia. Parametric contrastive learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 715–724, 2021. 14, 106

[114] Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *International Conference on Machine Learning*, pages 819–827. PMLR, 2013. 14

[115] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016. 16, 53, 54, 56, 64, 68, 85, 102, 106

[116] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018. 16, 53, 54, 56, 64, 66, 68

[117] Rui Huang, Andrew Geng, and Yixuan Li. On the importance of gradients for detecting distributional shifts in the wild. In *Advances in Neural Information Processing Systems*, volume 34, 2021. 16, 53, 54, 56, 64, 68

[118] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1563–1572, 2016. 16, 68

[119] Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. 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*, pages 10951–10960, 2020. 16, 55, 67, 68, 69

[120] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in Neural Information Processing Systems*, 31, 2018. 16, 53, 68

[121] Haoran Wang, Weitang Liu, Alex Bocchieri, and Yixuan Li. Can multi-label classification networks know what they don't know? *Advances in Neural Information Processing Systems*, 34, 2021. 16, 68

[122] Peyman Morteza and Yixuan Li. Provable guarantees for understanding out-of-distribution detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. 16, 68

[123] Yiyou Sun, Chuan Guo, and Yixuan Li. React: Out-of-distribution detection with rectified activations. *Advances in Neural Information Processing Systems*, 34, 2021. 16, 53, 68

[124] Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. In *International Conference on Machine Learning (ICML)*. PMLR, 2022. 16, 53, 68

[125] Zhaowei Zhu, Zihao Dong, and Yang Liu. Detecting corrupted labels without training a model to predict. In *International Conference on Machine Learning (ICML)*. PMLR, 2022. 16, 68

[126] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*, 2017. 16, 68, 107

[127] Petra Bevandić, Ivan Krešo, Marin Oršić, and Siniša Šegvić. Discriminative out-of-distribution detection for semantic segmentation. *arXiv preprint arXiv:1808.07703*, 2018.

[128] Yonatan Geifman and Ran El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. In *International Conference on Machine Learning*, pages 2151–2159. PMLR, 2019.

[129] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. *arXiv preprint arXiv:1802.10501*, 2018.

[130] Sina Mohseni, Mandar Pitale, JBS Yadawa, and Zhangyang Wang. Self-supervised learning for generalizable out-of-distribution detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5216–5223, 2020.

[131] Taewon Jeong and Heeyoung Kim. Ood-maml: Meta-learning for few-shot out-of-distribution detection and classification. *Advances in Neural Information Processing Systems*, 33, 2020.

[132] Jiefeng Chen, Yixuan Li, Xi Wu, Yingyu Liang, and Somesh Jha. Atom: Robustifying out-of-distribution detection using outlier mining. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 430–445. Springer, 2021.

[133] Hongxin Wei, Lue Tao, Renchunzi Xie, and Bo An. Open-set label noise can improve robustness against inherent label noise. In *Advances in Neural Information Processing Systems*, 2021. 72

[134] Hongxin Wei, Lue Tao, Renchunzi Xie, Lei Feng, and Bo An. Open-sampling: Exploring out-of-distribution data for re-balancing long-tailed datasets. In *International Conference on Machine Learning (ICML)*. PMLR, 2022.

[135] Yifei Ming, Ying Fan, and Yixuan Li. Poem: Out-of-distribution detection with posterior sampling. In *International Conference on Machine Learning (ICML)*. PMLR, 2022. 16, 68

[136] Xuefeng Du, Zhaoning Wang, Mu Cai, and Yixuan Li. Vos: Learning what you don't know by virtual outlier synthesis. In *Proceedings of the International Conference on Learning Representations*, 2022.

[137] Xuefeng Du, Xin Wang, Gabriel Gozum, and Yixuan Li. Unknown-aware object detection: Learning what you don't know from videos in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[138] Julian Katz-Samuels, Julia Nakhleh, Robert Nowak, and Yixuan Li. Training ood detectors in their natural habitats. In *International Conference on Machine Learning (ICML)*. PMLR, 2022. 16, 68

[139] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. In *Advances in Neural Information Processing Systems*, 2020. 16, 68

[140] Vikash Sehwag, Mung Chiang, and Prateek Mittal. Ssd: A unified framework for self-supervised outlier detection. In *International Conference on Learning Representations*, 2021.

[141] Yifei Ming, Yiyou Sun, Ousmane Dia, and Yixuan Li. Cider: Exploiting hyperspherical embeddings for out-of-distribution detection. *arXiv preprint arXiv:2203.04450*, 2022. 16, 68

[142] H. Bao, G. Niu, and M. Sugiyama. Classification from pairwise similarity and unlabeled data. In *International Conference on Machine Learning*, pages 452–461, 2018. 17

[143] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. *Semi-Supervised Learning*. MIT Press, 2006.

[144] M. C. du Plessis, G. Niu, and M. Sugiyama. Analysis of learning from positive and unlabeled data. In *Advances in Neural Information Processing Systems*, pages 703–711, 2014.

[145] Lei Feng and Bo An. Leveraging latent label distributions for partial label learning. In *International Joint Conferences on Artificial Intelligence*, pages 2107–2113, 2018.

[146] Lei Feng and Bo An. Partial label learning by semantic difference maximization. In *International Joint Conferences on Artificial Intelligence*, pages 2294–2300, 2019.

[147] Lei Feng and Bo An. Partial label learning with self-guided retraining. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3542–3549, 2019.

[148] Chen Gong, Hengmin Zhang, Jian Yang, and Dacheng Tao. Learning with inadequate and incorrect supervision. In *2017 IEEE International Conference on Data Mining)*, pages 889–894. IEEE, 2017. 17

[149] Aditya Menon, Brendan Van Rooyen, Cheng Soon Ong, and Bob Williamson. Learning from corrupted binary labels via class-probability estimation. In *International Conference on Machine Learning*, pages 125–134, 2015. 17

[150] Tyler Sanderson and Clayton Scott. Class proportion estimation with application to multiclass anomaly rejection. In *Artificial Intelligence and Statistics*, pages 850–858, 2014. 17

[151] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh Annual Conference on Computational Learning Theory*, pages 92–100, 1998. 18, 20

[152] Abhishek Kumar, Avishek Saha, and Hal Daume. Co-regularization based semi-supervised domain adaptation. In *Advances in Neural Information Processing Systems*, pages 478–486, 2010. 32, 109

[153] Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin. A co-regularization approach to semi-supervised learning with multiple views. In *Proceedings of ICML Workshop on Learning With Multiple Views*, pages 74–79, 2005. 20, 32, 109

[154] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4320–4328, 2018. 18, 20

[155] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2691–2699, 2015. 23, 82, 84

[156] Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *Proceedings of the 5th International Conference on Learning Representation*, 2016. 23, 39

[157] Ryuichi Kiryo, Gang Niu, Marthinus C du Plessis, and Masashi Sugiyama. Positive-unlabeled learning with non-negative risk estimator. In *Advances in Neural Information Processing Systems*, pages 1675–1685, 2017. 23

[158] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Learning to learn from noisy labeled data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5051–5059, 2019. 23, 24

[159] Kun Yi and Jianxin Wu. Probabilistic end-to-end noise correction for learning with noisy labels. *arXiv preprint arXiv:1903.07788*, 2019. 23

[160] Brendan Van Rooyen, Aditya Menon, and Robert C Williamson. Learning with symmetric label noise: The importance of being unhinged. In *Advances in Neural Information Processing Systems*, pages 10–18, 2015. 24, 71

[161] Xiyu Yu, Tongliang Liu, Mingming Gong, Kayhan Batmanghelich, and Dacheng Tao. An efficient and provable approach for mixture proportion estimation using linear independence assumption. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4480–4489, 2018. 26

[162] Melody Y Guan, Varun Gulshan, Andrew M Dai, and Geoffrey E Hinton. Who said what: Modeling individual labelers improves classification. In *Thirty-Second AAAI Conference on Artificial Intelligence*, pages 3109–3118, 2018. 33, 45

[163] Shadi Albarqouni, Christoph Baur, Felix Achilles, Vasileios Belagiannis, Stefanie Demirci, and Nassir Navab. Aggnet: Deep learning from crowds for mitosis detection in breast cancer histology images. *IEEE Transactions on Medical Imaging*, 35(5):1313–1321, 2016. 33

[164] Junbing Li, Changqing Zhang, Joey Tianyi Zhou, Huazhu Fu, Shuyin Xia, and Qinghua Hu. Deep-lift: Deep label-specific feature learning for image annotation. *IEEE Transactions on Cybernetics*, pages 1–10, 2021. doi: 10.1109/TCYB.2021.3049630. 33

[165] Filipe Rodrigues, Francisco Pereira, and Bernardete Ribeiro. Learning from multiple annotators: Distinguishing good from random labelers. *Pattern Recognition Letters*, 34(12):1428–1436, 2013. 33, 45

[166] Zhi-Hua Zhou. *Ensemble Methods: Foundations and Algorithms*. CRC press, 2012. 33, 45

[167] Bo Han, Ivor W. Tsang, Ling Chen, Joey Tianyi Zhou, and Celina P. Yu. Beyond majority voting: A coarse-to-fine label filtration for heavily noisy labels. *IEEE Transactions on Neural Networks and Learning Systems*, 30 (12):3774–3787, 2019. doi: 10.1109/TNNLS.2019.2899045. 33

[168] Hongxin Wei, Renchunzi Xie, Lei Feng, Bo Han, and Bo An. Deep learning from multiple noisy annotators as a union. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 34

[169] Erich L Lehmann and George Casella. *Theory of Point Estimation*. Springer Science & Business Media, 2006. 38

[170] Yivan Zhang, Nontawat Charoenphakdee, and Masashi Sugiyama. Learning from indirect observations. *arXiv preprint arXiv:1910.04394*, 2019. 38

[171] Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah M Erfani, Shu-Tao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. *arXiv preprint arXiv:1806.02612*, 2018. 38

[172] Bo Han, Jiangchao Yao, Gang Niu, Mingyuan Zhou, Ivor Tsang, Ya Zhang, and Masashi Sugiyama. Masking: A new perspective of noisy supervision. In *Advances in Neural Information Processing Systems*, pages 5836–5846, 2018.

[173] Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. Are anchor points really indispensable in label-noise learning? In *Advances in Neural Information Processing Systems*, pages 6835–6846, 2019. 42

[174] Hongxin Wei, Lue Tao, Renchunzi Xie, and Bo An. Open-set label noise can improve robustness against inherent label noise. In *Thirty-Fifth Conference on Neural Information Processing Systems*, pages 7978–7992, 2021.

[175] Yazhou Yao, Zeren Sun, Chuanyi Zhang, Fumin Shen, Qi Wu, Jian Zhang, and Zhenmin Tang. Jo-src: A contrastive approach for combating noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5192–5201, 2021.

[176] Zeren Sun, Xian-Sheng Hua, Yazhou Yao, Xiu-Shen Wei, Guosheng Hu, and Jian Zhang. Crssc: salvage reusable samples from noisy data for robust learning. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 92–101, 2020.

[177] Zeren Sun, Yazhou Yao, Xiu-Shen Wei, Yongshun Zhang, Fumin Shen, Jianxin Wu, Jian Zhang, and Heng Tao Shen. Webly supervised fine-grained recognition: Benchmark datasets and an approach. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10602–10611, 2021.

[178] Jonathan Krause, Benjamin Sapp, Andrew Howard, Howard Zhou, Alexander Toshev, Tom Duerig, James Philbin, and Li Fei-Fei. The unreasonable effectiveness of noisy data for fine-grained recognition. In *European Conference on Computer Vision*, pages 301–320. Springer, 2016.

[179] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pages 2304–2313. PMLR, 2018.

[180] Peng Hu, Xi Peng, Hongyuan Zhu, Liangli Zhen, and Jie Lin. Learning cross-modal retrieval with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5403–5413, 2021.

[181] Mouxing Yang, Yunfan Li, Zhenyu Huang, Zitao Liu, Peng Hu, and Xi Peng. Partially view-aligned representation learning with noise-robust contrastive loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1134–1143, 2021. 38

[182] Peng Cao, Yilun Xu, Yuqing Kong, and Yizhou Wang. Max-mig: An information theoretic approach for joint learning from crowds. *arXiv preprint arXiv:1905.13436*, 2019. 41, 43

[183] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. MIT Press, 2012. 42, 115

[184] Nan Lu, Tianyi Zhang, Gang Niu, and Masashi Sugiyama. Mitigating overfitting in supervised classification from two unlabeled datasets: A consistent risk correction approach. In *AISTATS*, pages 1115–1125, 2020. 42

[185] Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. *arXiv preprint arXiv:1712.06541*, 2017. 42

[186] Shahar Mendelson. Lower bounds for the empirical minimization algorithm. *IEEE Transactions on Information Theory*, 54(8):3797–3803, 2008. 43

[187] Kyohei Atarashi, Satoshi Oyama, and Masahito Kurihara. Semi-supervised learning from crowds using deep generative models. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 1555–1562, 2018. 43

[188] Michael Buhrmester, Tracy Kwang, and Samuel D Gosling. Amazon's mechanical turk: A new source of inexpensive, yet high-quality data? 2016. 45

[189] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: A database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173, 2008. 45

[190] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 46

[191] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015. 53

[192] Chandramouli Shama Sastry and Sageev Oore. Detecting out-of-distribution examples with gram matrices. In *International Conference on Machine Learning*, pages 8491–8501, 2020. 53

[193] Hongxin Wei, Xie Renchunzi, Cheng Hao, Lei Feng, Bo An, and Li Yixuan. Mitigating neural network overconfidence with logit normalization. In *International Conference on Machine Learning (ICML)*. PMLR, 2022. 55, 106

[194] Wilhelm Forst and Dieter Hoffmann. *Optimization—Theory and Practice*. Springer Science & Business Media, 2010. 59, 65

[195] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014. 62, 85, 96

[196] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011. 62, 85, 96

[197] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 62, 85, 96

[198] Pingmei Xu, Krista A Ehinger, Yinda Zhang, Adam Finkelstein, Sanjeev R Kulkarni, and Jianxiong Xiao. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv preprint arXiv:1504.06755*, 2015. 62, 85, 96

[199] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035. 2019. 62

[200] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330, 2017. 65, 69

[201] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2901–2907, 2015. 65

[202] Engkarat Techapanurak and Takayuki Okatani. Hyperparameter-free out-of-distribution detection using softmax of scaled cosine similarity. *arXiv:1905.10628*, 2019. 67, 69

[203] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*, 2016. 68

[204] Zhirong Wu, Yuanjun Xiong, X Yu Stella, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[205] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748, 2018. 68

[206] Rajeev Ranjan, Carlos D Castillo, and Rama Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*, 2017. 68

[207] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 212–220, 2017. 68

[208] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1041–1049, 2017. 68

[209] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018. 68

[210] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[211] Xiao Zhang, Rui Zhao, Yu Qiao, Xiaogang Wang, and Hongsheng Li. Adacos: Adaptively scaling cosine logits for effectively learning deep face representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10823–10832, 2019. 68

[212] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 68

[213] Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin. Understanding and improving layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019. 68

[214] Simon Kornblith, Ting Chen, Honglak Lee, and Mohammad Norouzi. Why do better loss functions lead to less transferable features? In *Advances in Neural Information Processing Systems*, 2021. 69

[215] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3):61–74, 1999. 69

[216] Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *International Conference on Machine Learning*, pages 609–616, 2001. 69

[217] Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 69

[218] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017. 69

[219] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip HS Torr, and Puneet K Dokania. Calibrating deep neural networks using focal loss. 2020. 69

[220] Jing Lei, James Robins, and Larry Wasserman. Distribution-free prediction sets. *Journal of the American Statistical Association*, 108(501):278–287, 2013. 69

[221] Chirag Gupta and Aaditya Ramdas. Top-label calibration and multiclass-to-binary reductions. In *International Conference on Learning Representations*, 2022. 69

[222] Deng-Bao Wang, Lei Feng, and Min-Ling Zhang. Rethinking calibration of deep neural networks: Do not be afraid of overconfidence. In *Advances in Neural Information Processing Systems*, 2021. 69

[223] Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 233–242, 2017. 71, 72, 75, 76

[224] Pengfei Chen, Guangyong Chen, Junjie Ye, Jingwei Zhao, and Pheng-Ann Heng. Noise against noise: stochastic label noise helps combat inherent label noise. In *International Conference on Learning Representations*, 2021. 72, 77, 79, 80, 82

[225] Aristotelis-Angelos Papadopoulos, Mohammad Reza Rajati, Nazim Shaikh, and Jiamian Wang. Outlier exposure with confidence control for out-of-distribution detection. *Neurocomputing*, 441:138–150, 2021. 74

[226] Antonio Torralba, Rob Fergus, and William T Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11): 1958–1970, 2008. 75, 79, 82

[227] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020. 75, 79

[228] Preetum Nakkiran, Behnam Neyshabur, and Hanie Sedghi. The deep bootstrap: Good online learners are good offline generalizers. *arXiv preprint arXiv:2010.08127*, 2020. 75, 79

[229] Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012. 78, 80

[230] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

[231] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

[232] Bobby Kleinberg, Yuanzhi Li, and Yang Yuan. An alternative view: When does sgd escape local minima? In *Proceedings of International Conference on Machine Learning*, pages 2698–2707. PMLR, 2018.

[233] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring generalization in deep learning. *arXiv preprint arXiv:1706.08947*, 2017. 78, 80

[234] Lingxi Xie, Jingdong Wang, Zhen Wei, Meng Wang, and Qi Tian. Disturblabel: Regularizing cnn on the loss layer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4753–4762, 2016. 79

[235] Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. *Advances in Neural Information Processing Systems*, 33:7597–7610, 2020. 82

[236] Hongxin Wei, Lue Tao, Renchunzi Xie, Lei Feng, and Bo An. Open-sampling: Exploring out-of-distribution data for re-balancing long-tailed datasets. In *International Conference on Machine Learning*, pages 23615–23630. PMLR, 2022. 88

[237] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision*, December 2015. 88, 95

[238] Hongxin Wei, Lue Tao, Renchunzi Xie, and Bo An. Open-set label noise can improve robustness against inherent label noise. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 90, 94, 104, 106

[239] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, 2010. 96

[240] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015. 97

[241] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 97

[242] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 97

[243] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 11662–11671, 2020. 97, 99

[244] Nikos Komodakis and Spyros Gidaris. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018. 97

[245] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8026–8037, 2019. 97

[246] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4004–4012, 2016. 101

[247] Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao. Range loss for deep face recognition with long-tailed training data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5409–5418, 2017. 101

[248] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017. 101

[249] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 106

[250] Saehyung Lee, Changhwa Park, Hyungyu Lee, Jihun Yi, Jonghyun Lee, and Sungroh Yoon. Removing undesirable feature contributions using out-of-distribution data. *arXiv preprint arXiv:2101.06639*, 2021. 106

[251] Hongxin Wei, Lei Feng, Rundong Wang, and Bo An. Metainfonet: Learning task-guided information for sample reweighting. *arXiv preprint arXiv:2012.05273*, 2020. 106

[252] Zhaowei Zhu, Tongliang Liu, and Yang Liu. A second-order approach to learning with instance-dependent label noise. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[253] Hao Cheng, Zhaowei Zhu, Xingyu Li, Yifei Gong, Xing Sun, and Yang Liu. Learning with instance-dependent label noise: a sample sieve approach. In *International Conference on Learning Representations*, 2021.

[254] Jingkang Wang, Hongyi Guo, Zhaowei Zhu, and Yang Liu. Policy learning using weak supervision. *Advances in Neural Information Processing Systems*, 34, 2021.

[255] Zhaowei Zhu, Yiwen Song, and Yang Liu. Clusterability as an alternative to anchor points when learning with noisy labels. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

[256] Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learning with noisy labels revisited: A study using real-world human annotations. In *International Conference on Learning Representations*, 2022.

[257] Zhaowei Zhu, Zihao Dong, and Yang Liu. Detecting corrupted labels without training a model to predict. In *International Conference on Machine Learning (ICML)*, 2022.

[258] Zhaowei Zhu, Jialu Wang, and Yang Liu. Beyond images: Label noise transition matrix estimation for tasks with lower-quality features. In *International Conference on Machine Learning (ICML)*, 2022. 106

[259] Jong-Chyi Su, Zezhou Cheng, and Subhransu Maji. A realistic evaluation of semi-supervised learning for fine-grained classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12966–12975, 2021. 106

[260] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. 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*, pages 41–50, 2019. 106

[261] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 107

[262] Tom Brown et al. Language models are few-shot learners. *NeurIPS*, 2020. 111

[263] Jacob Devlin et al. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 111

[264] Alec Radford et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 111

[265] Andreas Maurer. A vector-contraction inequality for rademacher complexities. In *International Conference on Algorithmic Learning Theory*, pages 3–17. Springer, 2016. 115