8-31-2022 11:00 AM

Understanding Deep Learning with Noisy Labels

Li Yi, *The University of Western Ontario*

Supervisor: McLeod, A. Ian, *The University of Western Ontario*
Co-Supervisor: Wang, Boyu, *The University of Western Ontario*
A thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Statistics and Actuarial Sciences

Abstract

Over the past decades, deep neural networks have achieved unprecedented success in image classification, which largely relies on the availability of correctly annotated large-scale datasets. However, collecting high-quality labels for large-scale datasets is expensive and time-consuming or even infeasible in practice. Approaches to addressing this issue include: acquiring labels from non-expert labelers, crowdsourcing-like platforms or other unreliable resources, where the label noise is inevitably involved. It becomes crucial to develop methods that are robust to label noise.

In this thesis, we study *deep learning with noisy labels from two aspects*. Specifically, the first part of this thesis, including two chapters, is devoted to learning and understanding representations of data with respect to label noise. In Chapter 2, we propose a novel regularization function to learn noise-robust representations of data such that classifiers are more reluctant to memorize the label noise. By theoretically investigating the representations induced by the proposed regularization function, we reveal that the learned representations keep information related to true labels and discard information related to corrupted labels, which indicates the robustness of the learned representations. Unlike Chapter 2 which leverages noisy labels, Chapter 3 studies representation learning without leveraging any label information, termed as self-supervised representations, and focuses on a more realistic scenario where the label noise is instance-dependent. From both theoretical analysis and empirical results, we show that the self-supervised representations have two benefits: (1) the instance-dependent label noise uniformly spreads over the representations; (2) the representations exhibit an intrinsic cluster structure with respect to true labels. The benefits encourage learned classifiers to be aligned better with the optimal classifiers.

The second part is devoted to understanding the connection between source-free domain adaptation (SFDA) and learning with noisy labels. In Chapter 4, we study SFDA from the perspective of learning with noisy labels and show that SFDA can be formulated as noisy label problems. In particular, we theoretically show that one fundamental challenge in SFDA is that the label noise is unbounded, which violates the basic assumption in conventional label noise scenarios. Consequently, we also show that the label noise methods based on noise-robust loss functions are not able to address it. On the other hand, we prove that the early-time training phenomenon exists in unbounded label noise scenarios. We conduct extensive experiments to demonstrate significant improvements to existing SFDA algorithms by leveraging the phenomenon.

Keywords: Representation learning, noisy labels, self-supervised learning, label noise

Summary for Lay Audience

Over the past decade, deep supervised learning has demonstrated its success in many areas, such as face detection, medical diagnosis, weather forecasting, customer discovery, etc. The success of deep supervised learning is primarily due to correct annotations of large-scale datasets. Existing algorithms of deep supervised learning are very sensitive to the reliabilities of annotations. Incorrect annotations will significantly affect the performance of these algorithms. False relationships might be captured when there are incorrect annotations in datasets. Collecting reliable annotations is time-consuming and expensive, so unreliable annotations are pervasive in many datasets. Therefore, the purpose of this research is to understand these unreliable annotations and build algorithms to prevent from obtaining false relationships.

In this thesis, we show that our algorithms can successfully avoid negative effects from unreliable annotations and we provide theoretical justifications for them. The benefits of these algorithms can be applied to various fields such as medical image diagnosis and autonomous driving. Both of these fields are likely to contain unreliable annotations from the human. Applying our algorithms can significantly reduce time and economic savings since collecting pure clean annotations for data is no longer needed. We conduct extensive experiments to show the effectiveness of our algorithms.

Co-Authorship Statement

This thesis is mainly based on three research articles.

The first part of the thesis (Chapter 2) has been published in Computer Vision and Pattern Recognition Conference (CVPR) 2022.

On Learning Contrastive Representations for Learning with Noisy Labels

Li Yi, Sheng Liu, Qi She, Ian McLeod, Boyu Wang

I am the main contributor to this work. Sheng Liu contributed to discuss the possible directions for loss and framework designs. Qi She helps me check grammatical errors and polish the language. I perform all necessary experiment design, analysis of results, and writing under the supervision of Dr. McLeod and Dr. Wang.

The second part of the thesis (Chapter 3) has been submitted in International Conference on Machine Learning on January.

How Self-supervised Pretrained Model Helps Learning with Label Noise

Li Yi, Changjian Shui, Ian McLeod, and Boyu Wang

I am the main contributor to this work. Changjian Shui helps me check grammatical errors and polish the language. I perform all necessary experiment design, analysis of results, and writing under the supervision of Dr. McLeod and Dr. Wang.

The third part of the thesis (Chapter 4) has been submitted in Neural Information Processing Systems on May.

When Source-Free Domain Adaptation Meets Learning with Noisy Labels

Li Yi, Pengcheng Xu, Ian McLeod, and Boyu Wang

I am the main contributor to this work. Pengcheng Xu helps me in reviewing related work for source-free domain adaptation. I perform all necessary experiment design, analysis of results, and writing under the supervision of Dr. McLeod and Dr. Wang.

I am grateful to acknowledge my supervisors Dr. Ian McLeod and Dr. Boyu Wang for their contributions to these articles.

Acknowlegements

First and foremost, I would like to express my special appreciation to my supervisor Dr. Ian McLeod for his constant support during my whole four-year Ph.D. journey at Western. Ian gives me the freedom that allows me to pursue interesting research topics and provides a respectful culture from which I truly benefited. I am also very lucky to be co-supervised by Dr. Boyu Wang. He is open-minded and has a great sense of research taste. I would like to express my sincere thanks to Boyu for his invaluable advice and constructive comments on my research. This thesis could never be completed without their mentorship.

I would like to thank my former co-supervisor Dr. Jiandong Ren for his supervision during my first-year Ph.D. I benefit a lot from discussing reinforcement learning with Jiandong, which lays a solid foundation for my further studies on deep learning.

I would also like to thank my collaborators throughout my Ph.D.: Sheng Liu, Changjian Shui, and Pengcheng Xu. It is my privilege to be able to work with them and I benefit a lot from the inspiring discussions with them.

I would like to appreciate my thesis commitee members: Hao Yu, Grace Yi, Charles Ling, and Yi Yang. The detailed feedback and suggestions are valuable to make a consistent better and solid thesis.

I also want to thank my friends at Western: Junhe Chen, Shi Chang, Xin Gu, Yiming Huang, jiaqi Mu, Wenjun Jiang, Ang Li, Jiaqi Li, Yifan Li, Yuying Li, Yuanhao Lai, Yang Miao, Ruizhi Pu, Junquan Xiao, Yilin Xie, Yichen Zhu, Yishan Zhang and many others. Spending time with them makes my life at Western more colorful.

Last but most importantly, I would like to thank my parents for their selfless love and unconditional support. Besides my parents, I also want to thank my girlfriend who always encourages me and stands behind me when I have difficulties. This thesis is dedicated to all of you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Deep neural network has demonstrated its success in varies areas such as computer vision [57, 101], natural language processing [23, 47], speech recognition [34, 2], etc. However, the success of deep neural networks largely rely on the availability of correctly labeled large-scale data. For example, the ImageNet recognition data contain more than 14 million manually-annotated images [22] that are time-consuming and expensive to collect. In recent years, there are an increasing number of datasets that are annotated by machine and/or non-expert labelers with minimum human supervision. One possible solution to quickly obtain the large-scale labeled data is to collect them from the Internet and extract key words or the key surrounding text as labels. For example, Clothing-1M (about 50% noise) contains 1 million clothing images from online shopping websites and each image is automatically assigned with a noisy label according to the key words ([141]). Food-101 (about 20% noise) is another data containing 300K food images with automatically assigned labels and researchers manually refine 4K of them ([65]). WebVision contains more than 2M images crawled from the Flickr website and Google Images search with no human supervision ([70]). Therefore, label noise is pervasive and noise level varies across datasets.

During the last few decades, deep learning models have been adapted to help humans make decisions. However, due to the existence of label noise in training datasets, deep learning models may fail dramatically since the performance of deep learning models hinges on the quality of collected data [114]. Underperformed models can mislead humans to make correct decisions. The problem is more severe among tasks such as medical diagnosis, credit default risk, etc. For these tasks, a minor mistake will be fatal to patients or society. Thus, it is crucial to develop algorithms that are able to address issues of label noise.

This thesis is centered around understanding representation learning for noisy data and a non-trivial application of label noise methods.

Representation Learning for Noisy Data

As our world has undergone drastic changes with the emergence of ”big data”, both the size of the dimension of have reached a large scale. This is the case in genetics where millions of genes are measured for a single individual, computer vision where a high resolution image contains hundreds of thousands of pixels, and so on. Representation learning is intended to map high-dimensional data to low-dimensions, making the model easier to discover patterns

of each class. Learning and understanding representations of data achieves great progress in supervised learning with clean labels, supervised learning with imbalanced datasets, supervised knowledge distillation [55, 73, 124], and etc. However, it is not well-understood whether and how representation learning can also benefit learning with noisy labels. Chapter 2 and Chapter 3 are devoted to learning and understanding representations of data with respect to label noise. Since labels of data are corrupted, the learned data representations are no longer reliable. A correctly labeled example and a mislabeled example are will mapped to nearby locations if their annotated labels are the same.

In Chapter 2, we propose to learn representations that examples from the same category are mapped to nearby locations and examples from different categories are mapped to distant locations. With this representation structure, a linear classifier can hardly memorize mislabeled examples, which is good for training reliable classifiers. We theoretically justified the optimal representations learned by maximizing mutual information with respect to features and inputs are robust to label noise. As indicated in [85] that some loss functions are robust to label noise but also suffer from underfitting problems. To this end, we also prove that the learned representations contain sufficient ground-truth label-related information to avoid underfitting. As the existence of label noise, we cannot obtain the optimal representations above. To overcome the effects of label noise, we propose a novel and gradient motivated algorithm that can effectively learn representations with the help of pseudo labels.

In Chapter 3, our work focuses on the self-supervised representation learning without any label information, whereas Chapter 2 focuses on the supervised representation learning with the help of noisy labels. Besides that, in Chapter 3, we extend the label noise from instance-independent to instance-dependent label noise, which is a more realistic label noise assumption. We reveal that even without using label information to guide representation learning, learning with label noise can still benefit from the learned self-supervised data representations. To understand it, we construct a motivating example of instance-dependent label noise, and theoretically find label noise are randomly distributed over the learned self-supervised representations, making the label noise easier to be addressed. The underlying reason for this is that the self-supervised representations can capture discriminative features of data, where label noise is assumed to correlated to non-discriminative features. On the other hand, we prove that self-supervised representations exhibit a good cluster structure that encourages the linear classifiers aligned better with ground-truth classifiers in the presence of noisy labels.

Connecting Source-Free Domain Adaptation to Learning With Label Noise

Though collecting clean labels for large-scale unlabeled training data is prohibitively expensive, collecting noisy labels (either from the Internet or from non-expert annotators) for them is still not affordable. In this case, one can resort to utilize the knowledge from other domains to help predicting unlabeled data. For example, a university email system (model) is used to prevent students from receiving junk emails tailored to them. To train such a model, a lot of educational emails need to be labeled, which is extremely expensive. A feasible solution is to buy an industrial spam email detection model to make it adapt to unlabeled educational emails. Due to the potential distribution shift between the industrial emails and the educational emails, directly deploying industrial model to the university email system without adaptation causes large prediction error. Formally, this problem is named source-free domain adaptation. Source-

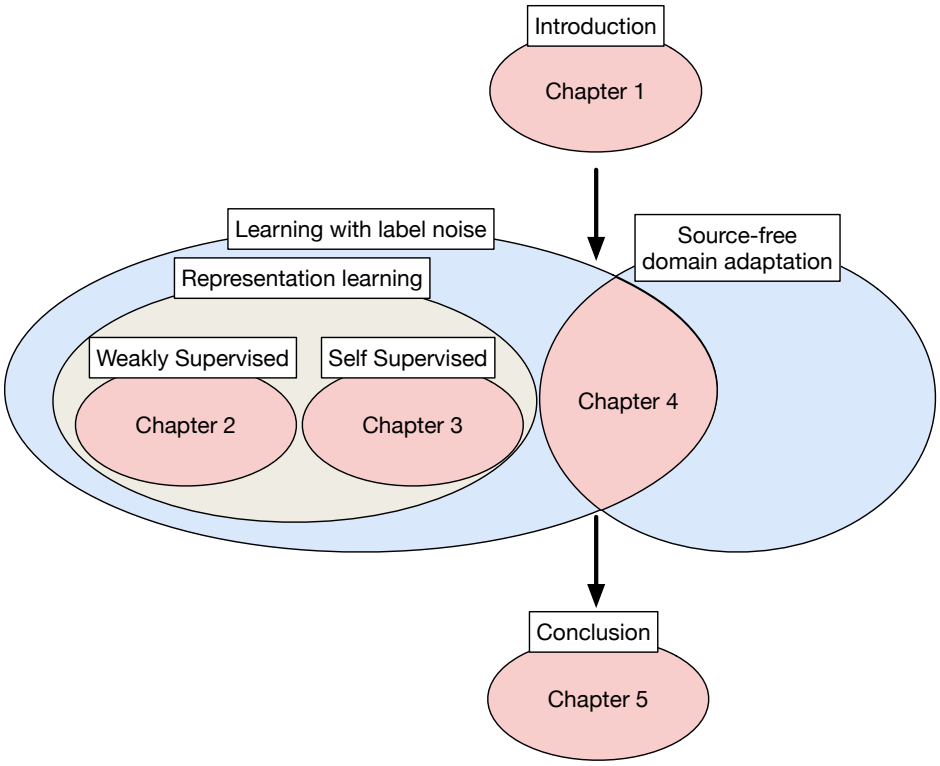


Figure 1.1: Thesis Structure

free domain adaptation has been explored by many empirical methods such as regularizing the cluster structure of unlabeled data or generating labeled images which have similar style to unlabeled data. A theoretical understanding of this problem is quite limited.

Chapter 4 aims to understanding the source-free domain adaptation problem through a perspective of learning with label noise. Theoretically, we show that the source-free domain adaptation problems can be formulated as the problems of learning with label noise. Unlike the label noise studied in Chapter 2 and Chapter 3, we prove that the label noise in source-free domain adaptation (Chapter 4) does not meet the assumptions of conventional label noise. Due to the discrepancy, many traditional label noise methods fail to generalize well to this problem. However, under the new assumption, we prove that the label noise can be addressed by leveraging early predictions. With and by leveraging this principle insights, existing source-free domain adaptation methods can be further boosted over various benchmark datasets. More precisely, by addressing the label noise existing in source-free domain adaptation, significant improvements can be achieved.

The outline of the thesis is illustrated in Figure 1.1. Chapter 1 introduces the foundations and background of learning with noisy labels. Chapter 2 proposes to learning representations with noisy labels. Chapter 3 focuses on learning representations without label information. Chapter 4 presents an understanding and connection between learning with noisy labels and source-free domain adaptation. Chapter 5 shows the conclusion of the thesis and the future research. The rest of the introductory chapter will include a part about related work.

1.1 Literature Review

In this section, we first provide an in-depth review of existing approaches for learning with label noise, and then we introduce the background of the contrastive learning which related to our method.

We use uppercases X, Y to represent input and output random variables, calligraphic letters $\mathcal{X}, \mathcal{Y}$ to represent sample spaces. We use X to represent inputs,Y to represent clean labels, and $\widetilde{Y}$ represent be noisy labels (which could be correct or incorrect). Let $f : \mathcal{X} \to \mathcal{Y}$ be a neural network with parameters θ and we omit the parameter θ for simplicity when the context is clear. We denote $\widetilde{\mathcal{D}}$ by the noisy training data which consists of some mislabeled samples, $\mathcal{D}$ by the clean data, and $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ by the objective function to minimize. The object is to use f to predict labels for unseen test data. When the training data is noise-free, the learned f usually performs well on the test data. When the training data is noisy, the performance of the classifier f on the test data drops owing to the fact that the mismatch between the training distribution $\widetilde{\mathcal{D}}$ and the test distribution $\mathcal{D}$ [35, 119].

Before we introduce approaches to solve noisy labels, we first introduce several noise types that have been studied in the literature of learning with noisy labels.

Instance-independent label noise. This is the most commonly studied type of label noise. It assumes $\Pr[\widetilde{Y} = j|Y = i, X = x] = \Pr[\widetilde{Y} = j|Y = i]$ or $\Pr[\widetilde{Y} = j|Y = i, X = x] = r$, where $Y, \widetilde{Y}$ are discrete. The former is the form of class-dependent label noise or asymmetric label noise, and the latter is the form of random label noise or symmetric label noise. Intuitively, the first one assumes that the corruptions exist between semantically-similar classes (e.g. dog and car, or bird and airplane), while the second one simply assumes that the probability of corruption is the same for every sample.

Instance-dependent label noise. Unlike instance-independent label noise, instance-dependent label noise is more realistic and states that the label corruption is dependent of X. However, without additional assumption, theoretically analyzing the instance-dependent label noise is impossible. Different samples could follow different label corruption distributions, which makes it hard to model these distributions since only one data point is available for each distribution.

1.1.1 Robust Regularization

For the deep learning domain, the deep neural networks are over parameterized, where the number of parameters is larger than the number of training samples. To avoid the overfitting problem, various regularization techniques have been implemented in neural networks. At first, regularization techniques are not proposed to solve noisy label issues but to improve the generalization ability. [156] shows that simple regularization techniques such as data augmentation, weight decay [58], dropout [118], and batch normalization [50] are also helpful for reducing memorization of noisy labels. However, their effects are only limited to a small proportion of noisy labels. More advanced regularization techniques are proposed for solving the effects of noisy labels.

[53] proposes an aggressive dropout regularization to encourage the neural network to learn a noise model, while [17] proposes a nested dropout to keep meaningful representations and

drop meaningless representations. [32] provides analysis about why the dropout makes the neural networks perform better.

Label smoothing [98] is another commonly used regularization method. In conventional classification tasks, the class label Y is represented as a one-hot vector. Label smoothing simply mixes the one-hot vector with the uniform label vector. In particular:

$$Y_{\text{LS}} = \alpha Y + (1 - \alpha) Y_U,$$

where Y is the original one-hot vector, $Y_U = (1, 1, \ldots, 1)^\top$, and α is a hyperparameter. A recent study [82] shows that label smoothing is beneficial to learning with noisy labels, while [136] justifies that the benefit of label smoothing vanishes when the noise level goes up. Meanwhile, [11] proposes a method called SLN, which adds Gaussian random variables to labels. Specifically:

$$Y_{\text{SLN}} = Y + \sigma Z_Y,$$

where σ is a hyperparameter and $Z_Y \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{|\mathcal{Y}|\times|\mathcal{Y}|})$. Similarly, [40] adds Gaussian random variables to the outputs $f(X)$. From a high-level point of view, label smoothing and its variants avoid overconfidence in label noise.

Both Y_{LS} and Y_{SLN} change the original label space from discrete to continuous, but this wound not change the space of the output variable. Given the cross-entropy loss $-Y^\top \log f(X)$, changing Y is simply as changing the weight for minimizing the loss between the Y and the probabilistic prediction for X.

Data augmentation such as mixup [157] has also been found helpful to reduce memorization of noisy labels. It is a linear interpolation between examples-labels pairs:

$$\widetilde{\mathcal{D}}_{\text{mix}} = \{(\bar{x}, \bar{y}) | \bar{x} = \lambda x_i + (1 - \lambda) x_j, \bar{y} = \lambda y_i + (1 - \lambda) y_j\},$$

where $(x_i, y_j), (x_j, y_j)$ are from the original dataset $\widetilde{\mathcal{D}}$, $\lambda \sim \text{Beta}(\alpha, \alpha)$, and α is a hyperparameter. Like label smoothing but more powerful than label smoothing, mixup can be easily incorporated into any framework, and it has been used in many label noise methods to further boost the performance [66, 90, 18, 75, 3, 152].

Early stopping has been theoretically proven to be robust to label noise when the noise level is low [69]. It serves as an implicit regularization method that stops the training progress before the training converges. [5] proposes to progressively stop training layers of neural networks from former layers to latter layers, while conventional early stopping stops training all layers simultaneously. Similar to the idea of early stopping, [138] divides parameters of neural networks into two groups: critical parameters and non-critical parameters. Then [138] only allows training for critical parameters and stops training for non-critical parameters. [48] regularizes the whole parameters of neural networks by penalizing the distance between them to their initial values, which is provably robust to noisy labels.

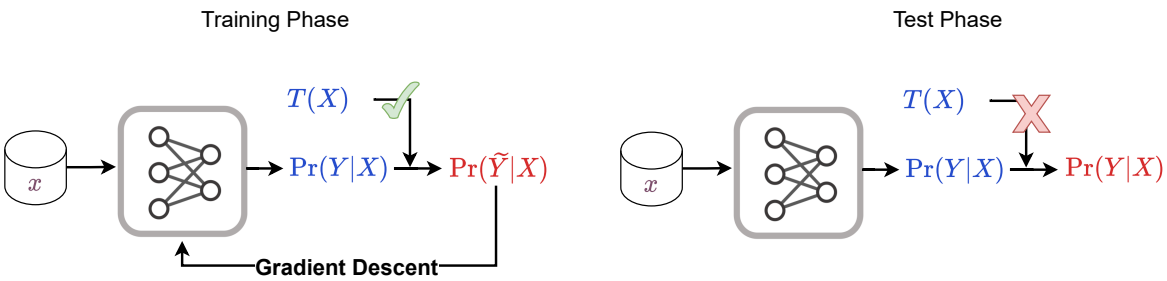


Figure 1.2: An illustration of loss correction method. During the training phase, we plug in the noise transition matrix $T(X)$ to the outputs of neural networks and make the model learn the correct probability $\Pr[Y|X]$. During the test phase, $T(X)$ is not embedded into the outputs of neural networks.

1.1.2 Loss Correction

The principle behind the loss correction methods is the following equation

$$\begin{aligned}\Pr[\tilde{Y}=j|X] &= \sum_i \Pr[\tilde{Y}=j, Y=i|X] \\ &= \sum_i \underbrace{\Pr[\tilde{Y}=j|Y=i,X]}_{\text{noise transition matrix}} \underbrace{\Pr[Y=i|X]}_{\text{base model}}.\end{aligned}$$

We let $T(X)_{ij} = \Pr[\tilde{Y}=j|Y=i,X]$ for simplicity, and $T(X) \in \mathbb{R}^{|\mathcal{Y}|\times|\mathcal{Y}|}$ is the transition matrix for sample X. The high-level idea of the loss correction method is illustrated in Figure 1.2.

Since the noisy conditional probability can be disentangled into two terms: the correct conditional probability and the noise transition matrix, we can model them independently. As Figure 1.2 shows, the neural network is used to model the correct conditional probability $\Pr[Y|X]$, and additional effects to estimate the noise transition matrix. [95] terms this the forward correction, and shows the theoretical result that the minimizer of the modified loss $\mathbb{E}_{\widetilde{D}}\mathcal{L}(T(X)^\top f(X), \tilde{Y})$ is equivalent to the minimizer of $\mathbb{E}_D\mathcal{L}(f(X), Y)$. [95] also proposes backward correction but empirically it performs worse than the forward correction.

Since the ground-truth labels are not given, directly calculating the noise transition matrix is impossible. Some work focus on estimating the noise transition matrix when $T(X)$ is independent of sample X. [92] focuses on binary classification and tunes the noise transition matrix by cross-validation. [76, 95] assumes that anchor examples exist, which means that $\forall j \in \mathcal{Y}, \exists x_{0j} \in \mathcal{X}, \Pr[Y=j|X_{0j}] = 1$. Then

$$\begin{aligned}\Pr[\widetilde{Y}=k|X_{0j}] &= \sum_i T_{ik} \Pr[Y=i|X_{0j}] \\ &= T_{jk},\end{aligned}$$

where in practice, the anchor examples are obtained by $x_{0j} = \arg\max_x \hat{\Pr}[\widetilde{Y}=j|x]$ for each $j \in \mathcal{Y}$. It shows that the noisy conditional probability for anchor examples produced by neural networks can be used to estimate the noise transition matrix. This approach has been used a

lot to estimate the noise transition matrix. [44] leverages a small proportion of clean dataset to provide more accurate estimation of the noise transition matrix, but the small set of clean examples are usually unavailable. Anchor examples play an important role in estimating the noise transition matrix. On the one hand, [140] focuses on the scenario that there are no anchor examples, which can lead to a poorly estimated noise transition matrix, and proposes T-Revision to efficiently learn the transition matrix. On the other hand, [151] improves the estimation accuracy by factorizing the original transition matrix into the product of two easy-to-estimate transition matrices. All of these methods focus on learning instance-independent noise transition matrix T. [8] focuses on a more realistic case, where noise transition matrices are dependent with instance X, and proposes to use neural networks to estimate $T(X)$.

1.1.3 Label Correction

Label correction methods achieve great success in addressing the memorization of noisy labels. The high-level idea behind the label correction is to replace wrong labels with correct labels so that the neural networks can learn meaningful information without being affected by noisy labels.

[102] is the first to propose the idea of label correction by the following equation:

$$Y_{\text{corrected}} = \alpha\widetilde{Y} + (1-\alpha)\hat{Y},$$

where $\hat{Y}$ is the probabilistic output from neural networks, and α is a hyperparameter. [3] dynamically updates α by a beta mixture model, which is built upon the sample loss values. Small α is assigned to the sample if it has a larger loss value. [11] directly updates α by assigning normalized loss values instead of modeling it by a probabilistic model. [86] relates α to local intrinsic dimensionality (LID) where small α is assigned to the sample if it has a larger LID value. [162] updates α by leveraging a small set of clean samples.

Instead of using linear interpolation between predicted labels and original labels as pseudo labels, [121] maintains a neural network to correct labels for noisy datasets. A similar idea is also given by [152]. [66] first divides samples into two groups, where the first group is the correctly labeled group and another is the incorrectly labeled group. Samples from the second group are re-labeled by the neural networks.

Some label correction methods are related to the confidence of label predictions. [113] found that consistent label predictions are likely to be correctly predicted and leverages this observation to correct noisy labels. [160] uses a progressive label correction strategy that only purifies labels for confident examples. [75] leverages the moving average of probabilistic predictions of samples as pseudo labels by a regularization term.

1.1.4 Robust Loss Function

For conventional classification tasks, cross-entropy (CE) loss is widely used. When training distribution is equivalent to test distribution, the neural network f trained from training data can also perform well on test data. Since the training data can be noisy, the training distribution is not equivalent to the test distribution. The learned f on noisy training data by using CE loss can easily memorize noisy labels and cause overfitting problems, making it generalize poorly

on test data [4, 87, 75]. Therefore, the key idea behind the robust loss function is to design a new loss function that does not make f memorize noisy labels. Specifically:

$$\arg\min_f \mathbb{E}_{\widetilde{\mathcal{D}}}[\mathcal{L}(f(X), \widetilde{Y})] = \arg\min_f \mathbb{E}_{\mathcal{D}}[\mathcal{L}(f(X), Y)],$$

where $\mathcal{L}$ is a noise robust loss function. It states that the optimal classifier f minimizing the loss $\mathcal{L}$ over noisy training data $\widetilde{\mathcal{D}}$ also minimizes the loss $\mathcal{L}$ over clean test data $\mathcal{D}$.

Under the mild noise assumption such as symmetric label noise or asymmetric label noise assumption, [29] shows the mean absolute error (MAE) loss is noise robust:

$$\mathcal{L}_{\text{MAE}}(X, Y) = \|f(X) - Y\|_1 ,$$

where $\|\cdot\|_1$ is the L1 norm. Since MAE loss cannot extract useful information from complicated data, generalized cross entropy (GCE) [161] solves this problem by combining both MAE and CE loss via Box-Cox transformation:

$$\mathcal{L}_{\text{GCE}}(X, Y) = \frac{(1 - (f(X)^\top Y))^q}{q},$$

where $q \in (0, 1]$ is a hyperparameter. Similar to GCE, SL [134] also proposes a noise-robust loss:

$$\mathcal{L}_{\text{RCE}}(X, Y) = f(X)^\top \log Y,$$

where one entry of Y is 1 and the rest entries of Y are 0 but the author defined $\log 0 = 1e^{-4}$. and combines it with CE loss

$$\mathcal{L}_{\text{SL}}(X, Y) = \alpha Y^\top \log f(X) + \beta \mathcal{L}_{\text{RCE}}(X, Y),$$

where α, β are hyperparameters. As observed by [85], robust loss functions GCE and SL are only partially robust to noisy labels because of CE loss. [85] proposes to normalize CE as normalized CE loss is also robust to label noise. Unlike GCE, [24] proposes another robust loss function called GJS, which is another interpolation between CE and MAE. Specifically:

$$\mathcal{L}_{\text{GJS}}(X, Y) = \sum_{i=1}^{M} \pi_i D_{\text{KL}}(\boldsymbol{p}^{(i)}, \sum_{j=1}^{M} \pi_j \boldsymbol{p}^{(j)}),$$

where $M = 3$ in practice, $\pi_2 = \pi_3 = \frac{1-\pi_1}{2}$. We note that π_1 is the hyperparameter, $D_{\text{KL}}(p, q) = p^\top \log \frac{p}{q}$ is the KL divergence, $p^{(1)} = Y$, $p^{(2)} = f(X)$, $p^{(3)} = f(X^+)$, and X^+ is another view of sample X. [26] takes the Taylor series of CE loss, and states that by adjusting the order of the Taylor series, the adjusted CE loss can also be robust.

1.1.5 Sample Selection

We also discuss sample selection methods for learning with label noise. Here the sample selection also includes soft sample selection such as adding weights to different examples. We first discuss soft sample selection methods based on weights.

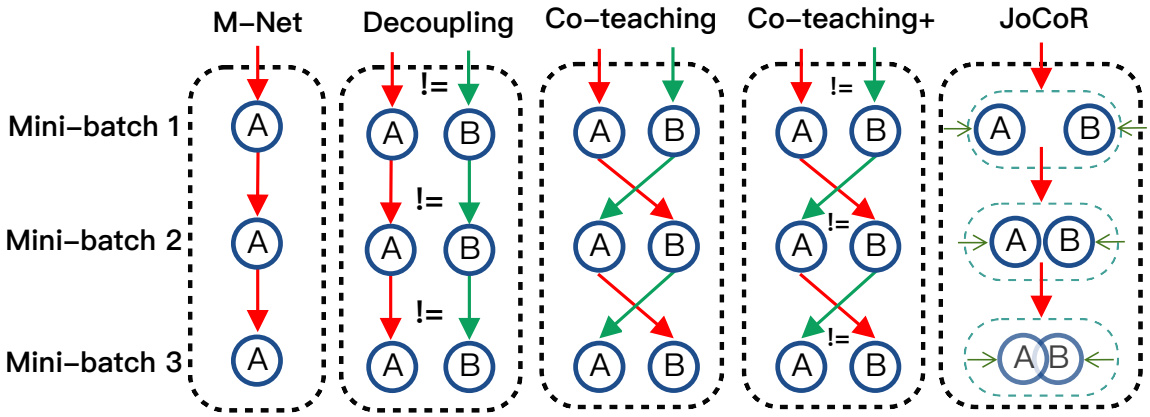


Figure 1.3: Selected sample flow for different sample selection methods, where A and B are denoted by the two different initialized neural networks. In each mini-batch data, each network will perform an update based on its updating strategy.

[76] uses importance weighting to learn the classifier f in the presence of noisy labels, which follows:

$$\mathbb{E}_{\mathcal{D}}[\mathcal{L}(f(X), Y)] = \mathbb{E}_{\widetilde{\mathcal{D}}}[\frac{P_{\mathcal{D}}(X, Y)}{P_{\widetilde{\mathcal{D}}}(X, \widetilde{Y})}\mathcal{L}(f(X), \widetilde{Y})].$$

By the above equation, minimizing the weighted loss on noisy training data is equivalent to minimizing the loss over clean data. [103] assigns weights w_i to the examples $(x_i, \tilde{y}_i)$, where the weights w_i is determined by a small set of clean samples. This method has also been used in [162]. [49] chooses to use the confidence score of model outputs as weights w_i.

For explicit sample selection methods, they aim to reduce the effects of mislabeled samples by only selecting clean data from noisy training data to update neural networks. [51] (M-Net) trains an extra neural network with a known set of clean samples and use the network to select clean samples from the noisy dataset. [88] (Decoupling) does not require extra clean data. Instead, Decoupling trains two neural networks, and the mislabeled samples are decided if the two neural networks give different predictions on these samples. [38] (Co-teaching) finds that small loss examples are usually clean examples. Co-teaching maintains two neural networks and mutually selects small loss examples to update the other neural network. While [154] (Co-teaching+) adjusts the selection strategy of Co-teaching method by only choosing small loss examples with different predictions given by two neural networks to update neural networks. [135] (JoCoR) proposes to update two neural networks simultaneously instead of iteratively. JoCoR reduces the diversity of two neural networks for better performance. The illustration of these methods is shown in Figure 1.3. The idea of selecting small loss examples as clean examples has been widely used. [113, 17] use the selection strategy the same Co-teaching. [66] uses Co-teaching strategy to separate noisy training data as a clean labeled part and an incorrectly labeled part.

In addition to these small loss selection methods, [83] proposes a Curriculum Loss and uses it as a criterion to automatically and adaptively select samples for training. Similarly, [18]

also designs a new selection criterion with theoretical guarantees. [90] finds that clean data can form a low-rank Jacobian matrix, and it selects clean samples which have the low-rank Jacobian.

1.1.6 Hybrid Approach

Most of above approaches can be jointly used with other approaches to obtain better performance. Though robust regularization only provides marginal improvement to models in the presence of noisy labels, it can be easily embedded into other methods to further boost model performance. For example, [75, 18, 90, 66, 3] all report when jointly using their methods with data augmentation technique mixup, the performance increases significantly. We also note that weight decay, as a regularization method, has already been automatically used in every modern neural network. [134] demonstrates the success when using label smoothing with a robust loss function SL. [11] also shows the success that adding random variables to labels together with simple label correction can bring significant improvement. Robust loss functions can also be jointly used with other methods such as label correction and loss correction, which are shown in [134, 153]. [113] shows the success of combining the sample selection method and label correction method.

1.1.7 Contrastive Learning

Contrastive learning is one of the representation learning approaches, which focuses on learning meaningful data representations. Different contrastive learning methods have different but similar objective functions to optimize [13, 14, 41, 91]. However, the high-level ideas of them are the same. For supervised contrastive learning, the goal is to learning representations from data, where representations from the same class forms a tight cluster, while clusters are pushed away from each other. A typical supervised contrastive learning framework is [55]. Given an image x_i and an image x_j from the same class. The supervised contrastive loss for this pair is defined as:

$$\mathcal{L}_{\text{supcon}} = -\log \frac{\exp\left(\text{sim}(x_i, x_j)/\tau\right)}{\sum_{k=1}^{N} \mathbb{1}\{k \neq i\} \exp\left(\text{sim}(x_i, x_k)/\tau\right)},$$

where $\text{sim}(x_i, x_j)$ is a cosine similarity function for the representations of x_i and x_j, τ is a hyperparameter. Minimizing this loss is equivalent to learn parameters of the representation model such that the data representations from the same class are pulled together, and data representations from different classes are pushed away.

As studied in [16, 36], by careful neural network frameworks design, the denominator of the above fraction can be discarded, which results into objective functions that only focus on regularizing representations from the same class. The pairs for them are termed positive pairs.

Chapter 2

On Learning Noise-Robust Representations for Learning with Label Noise

2.1 Introduction

The successes of deep neural networks [43, 104] largely rely on availability of correctly labeled large-scale datasets that are prohibitively expensive and time-consuming to collect [141]. Approaches to addressing this issue includes: acquiring labels from crowdsourcing-like platforms or non-expert labelers or other unreliable sources [145, 161] but while theses methods can reduce the labeling cost, label noise is inevitable. Due to the over-parameterization of deep networks [43], examples with noisy labels can ultimately be memorized with a cross entropy loss[75, 4, 90], which is known as the *memorization effect* [156, 87], leading to poor performance. Therefore, it is important to develop methods that are robust to the label noise.

Cross entropy (CE) loss is widely used as a loss function for image classification tasks due to its strong performance on clean training data [114] but it is not robust to label noise. When labels in training data are corrupted, the performance drops [6, 7]. Given the memorization effect of deep networks, training on noisy data with the CE loss results in the representations of the data clustered in terms of their noisy labels instead of the ground truth. Thus, the final layer of the deep networks cannot find a good decision boundary from these noisy representations.

To overcome the memorization effect, noise-robust loss functions have been actively studied in the literature [89, 161, 134, 26]. They aim to design noise-robust loss functions in a way such that they achieve small loss on clean data and large loss on wrongly labeled data. However, it has been empirically shown that being robust alone is not sufficient for a good performance as it also suffers from the *underfitting* problem [85]. To address this issue, these noise-robust loss functions have to be explicitly or implicitly jointly used with the CE loss, which brings a trade-off between non-robust loss and robust loss. As a result, the memorization effect is alleviated but still remains due to the non-robust CE loss.

In this chapter, we tackle this problem from a different perspective. Specifically, we investigate contrastive learning and the effect of the clustering structure for learning with noisy labels. Owing to the power of contrastive representation learning methods [16, 36, 14, 55, 15], learn-

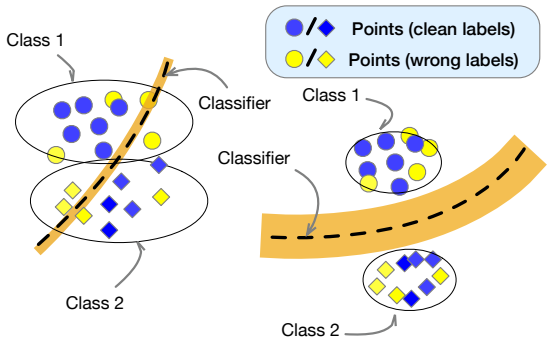


Figure 2.1: Illustration of the proposed method with noisy labels. Black curves are the best classifiers that are learned during training. **Left:** Deep networks without contrastive regularization. **Right:** Deep networks with contrastive regularization. Two classes are better separated by deep networks that points with the same class are pulled into a tight cluster and clusters are pushed away from each other.

ing contrastive representations has been extensively applied on various tasks [142, 117, 84]. The key component of contrastive learning is positive contrastive pair (x_1, x_2). Training a contrastive objective encourages the representations of x_1, x_2 to be closer. In classification tasks, correct positive contrastive pairs are formed by examples from the same class. When label noise exists, defining contrastive pairs in terms of their noisy labels results in adverse effects. Encouraging representations from different classes to be closer makes it even more difficult to separate images of different classes. Similar to our attempt to learn contrastive representations from noisy data, previous work has focused on reducing the adverse effects by re-defining contrastive pairs according to their pseudo labels [68, 19, 30, 67]. However, pseudo labels can be unreliable, and then wrong contrastive pairs are inevitable and can dominate the representation learning.

To address this issue, we propose a new contrastive regularization function that does not suffer from the adverse effects. We theoretically investigate benefits of representations induced by the proposed contrastive regularization function from two aspects. First, the representations of images keep information related to true labels and discard information related to corrupted labels. Second, we theoretically show that the classifier is hard to memorize corrupted labels given the learned representations, which demonstrates that our representations are robust to label noise. Intuitively, learning such contrastive representations of data helps combat the label noise. If data points are clustered tightly in terms of their true labels, then it makes the classifier hard to draw a decision boundary to separate the data in terms of their corrupted labels. We illustrate this intuition in Figure 2.1.

Our main contributions are as follows.

- We theoretically analyze the representations induced by the contrastive regularization function, showing that the representations keep information related to true labels and discard information related to corrupted labels. Moreover, we formally show that representations with insufficient corrupted label-related information are robust to label noise.
- We propose a novel algorithm over data with noisy labels to learn contrastive represen-

tations, and provide gradient analysis to show that correct contrastive pairs can dominate the representation learning.

- We empirically show that our method can be applied with existing label correction techniques and noise-robust loss functions to further boost the performance. We conduct extensive experiments to demonstrate the efficacy of our method.

2.2 Theoretical Analysis

In this section, we first introduce some notations and we then investigate the benefits of representations learned by the contrastive regularization function.

2.2.1 Preliminaries

We use uppercases $X, Y, \ldots$ to represent random variables, calligraphic letters $\mathcal{X}, \mathcal{Y}, \ldots$ to represent sample spaces, and lowercases $x, y, \ldots$ to represent their realizations. Let X be input random variable and Y be its true label. We use $\tilde{Y}$ to denote the wrongly-labeled random variable that is not equal to Y. The entropy of the random variable Y is denoted by $H(Y)$ and the mutual information of X and Y is $I(X, Y)$. We use $p(\cdot)$ to represent the density function and overload p_i as the probabilistic output for sample x_i.

Contrastive learning aims to learn representations of data that only the data from the same class have similar representations. In this chapter, we propose to learn the representations by introducing the following contrastive regularization function over all examples $\{(x_i, y_i)\}$ from $\mathcal{X} \times \mathcal{Y}$ and y_i is the ground truth.

$$\mathcal{L}_{\text{ctr}}(x_i, x_j) = -(\langle \tilde{q}_i, \tilde{z}_j \rangle + \langle \tilde{q}_j, \tilde{z}_i \rangle)\mathbb{1}\{y_i = y_j\}, \tag{2.1}$$

where $\tilde{q}_k = \frac{q_k}{\|q_k\|_2}$ and $\tilde{z}_k = \frac{z_k}{\|z_k\|_2}$, and x_i, x_j are inputs. Following SimSiam [16], we define $q = h(f(x))$, $z = \texttt{stopgrad}(f(x))$, f is an encoder network consisting of a backbone network and a projection MLP, and h is a prediction MLP. Minimizing Eq. (2.1) wish respect to the parameters of neural networks on $\{(x_i, y_i), (x_j, y_j)\}$ pulls representations of x_i and x_j closer if $y_i = y_j$. The designs of the stop-gradient operation and h applied on representations are mainly to avoid trivial constant solutions.

2.2.2 The Benefits of Representations Induced by Contrastive Regularization

We first relate the solutions that minimize Eq. (2.1) to a mutual information

$$I(Z; X^+) = \iint p(z, x^+) \log \frac{p(z|x^+)}{p(z)} \, \mathrm{d}x^+ \, \mathrm{d}z,$$

where $p(\cdot)$ is a density function, $z = f(x)$ and x^+ is from the same class as x. The mutual information characterizes the information of one random variable contained when given another random variable.

Theorem 2.2.1. *Representations Z learned by minimizing Eq. (2.1) maximizes the mutual information $I(Z; X^+)$, where the maximization is over the parameters of neural networks for calculating Z.*

The proof is provided in 2.B.1. Theorem 2.2.1 reveals the equivalence between the contrastive learning and mutual information maximization. Intuitively, Eq. (2.1) encourages to pull representations from the same class together and push those from different classes apart. The estimate of z conditioned on x^+ is more accurate than random guessing because the representation z of x is similar to the representation of x^+. Thus the pointwise mutual information $\log \frac{p(z|x^+)}{p(z)}$ increases by minimizing Eq. (2.1).

We denote $Z^\star = \arg\max_{Z_\theta} I(Z_\theta, X^+)$ by the representation that maximizes the mutual information, where Z_θ is a representation of X parameterized by the neural network f with parameters θ. To understand what $Z^\star$ is learned from inputs and to show that $Z^\star$ is noise-robust, we introduce the notion of (ϵ, γ)-*distribution*:

Definition 2.2.2 ((ϵ, γ)-distribution)**.** *A distribution $D(X, Y, \tilde{Y})$ is called (ϵ, γ)-Distribution if there exists $\gamma \gg \epsilon > 0$ such that*

$$I(X; Y|X^+) \leq \epsilon, \tag{2.2}$$

and

$$I(X; \tilde{Y}|X^+) > \gamma. \tag{2.3}$$

Eq. (2.2) characterizes the connection between images and their true labels. If we already know an image X^+, then there is the limited extra information related to the true label by additionally knowing X. We use a small number ϵ to restrict this additional information gain. Eq. (2.3) characterizes the connection between those images and their corrupted labels. By knowing an additional image X^+, the information X contains about its corrupted label $\tilde{Y}$ is still larger than γ. The above condition $\gamma \gg \epsilon > 0$ states that images from the same class are much more similar with respect to the true label than the corrupted label. As it is mentioned in [116], if there is a perfect prediction of Y given X^+, then $\epsilon = 0$.

We illustrate the intuitions behind Definition 2.2.2 in Figure 2.2. We use the Grad-CAM [109] to highlight the important regions in the images for predictions. The Grad-CAM maps the computed gradients to the original image and the magnitudes of the gradients highlight the important regions in the image. The highlighted regions captured by the model are most related to labels. For images with the same clean labels, their information related to true labels are similar. For example, when Cat 1 and Cat 2 in Figure 2.2 are labeled as "cat", cat faces are captured as the true label-related information and they all look alike. For images with corrupted labels, their information related to corrupted labels are quite different. When Cat 1 and Cat 2 in Figure 2.2 are labeled as "dog", the windows bars captured as the corrupted label-related information for Cat 1 is different from the floor and wall for Cat 2.

With the notion of (ϵ, γ)-distribution, the following theorem help us understand the benefits of representations $Z^\star$ in depth.

Theorem 2.2.3. *Given a distribution $D(X, Y, \tilde{Y})$ that is (ϵ, γ)-Distribution, we have*

$$I(X; Y) - \epsilon \leq I(Z^\star; Y) \leq I(X; Y), \tag{2.4}$$

$$I(Z^\star; \tilde{Y}) \leq I(X; \tilde{Y}) - \gamma + \epsilon. \tag{2.5}$$

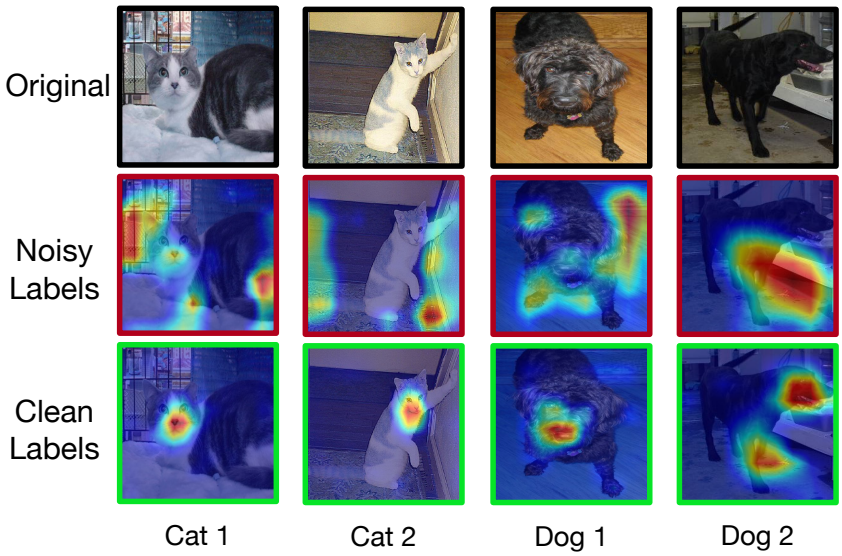


Figure 2.2: An example of Grad-CAM [109] results of ResNet34 trained on noisy dataset with 40% symmetric label noise and clean dataset, separately. When there is label noise, information related to corrupted labels captured by the model varies from image to image (e.g. window bars in Cat 1 v.s. floor and wall in Cat 2). When there is no label noise, information related to true labels are similar for images from the same class (e.g. cat face in Cat 1 v.s. cat face in Cat 2).

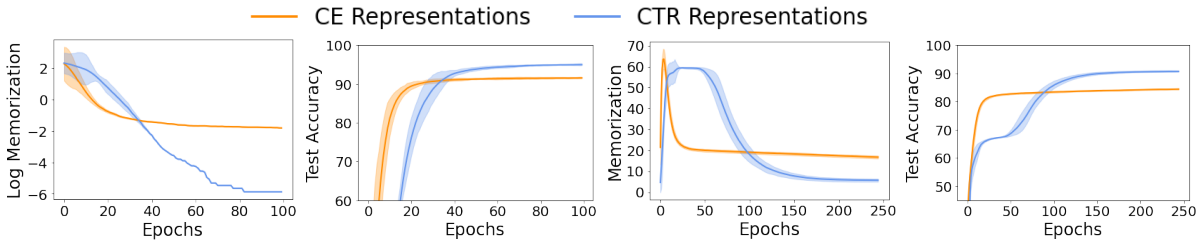
The proof is provided in 2.B.2. Given images X and their labels Y, the mutual information $I(X;Y)$ is fixed. The theorem states that the learned representations $Z^\star$ keep as much true label-related information as possible and discard much corrupted label-related information. Since the corrupted label-related information is discarded from the representations $Z^\star$, memorizing the corrupted labels based on $Z^\star$ is diminished. Lemma 2.2.4 establishes the lower bound on the expected error on *wrongly-labeled* data.

Lemma 2.2.4. *Consider a pair of random variables* $(X, \tilde{Y})$. *Let* $\hat{Y}$ *be outputs of any classifier based on inputs* Z_θ, *and* $\tilde{e} = \mathbb{1}\{\hat{Y} \neq \tilde{Y}\}$, *where* $\mathbb{1}\{A\}$ *be the indicator function of event* A. *Then, we have*

$$\mathbb{E}[\tilde{e}] \geq \frac{H(\tilde{Y}) - I(Z_\theta; \tilde{Y}) - H(\tilde{e})}{\log(|\tilde{\mathcal{Y}}|) - 1}.$$

The proof is provided in 2.B.3. Lemma 2.2.4 provides a necessary condition on the success of learning with noisy labels based on representation learning and sheds new light on this problem by highlighting the role of minimizing $I(Z_\theta; \tilde{Y})$. To see this, note that small $I(Z_\theta; \tilde{Y})$ implies robustness to label noise since $\mathbb{E}[\tilde{e}]$ is the expected error over the corrupted labels. On the other hand, when minimizing Eq. (2.1), small $I(Z^\star; \tilde{Y})$ can be achieved as indicated by the upper bound in Eq. (2.12). In the meanwhile, the lower bound on $I(Z^\star; Y)$ in Eq. (2.11) also shows that $Z^\star$ can retain the discriminative information of the data to avoid a trivial solution to $I(Z_\theta; \tilde{Y})$ minimization (i.e., Z_θ is a constant representation).

While Lemma 2.2.4 combined with Theorem 2.2.3 indicates that $Z^\star$ is robust to label noise, the following Lemma shows that $Z^\star$ can also avoid underfitting. Specifically, it implies that that a good classifier achieved under the clean distribution can also be achieved based on our representations $Z^\star$.



(a) 80% SYM (b) 80% SYM (c) 40% ASYM (d) 40% ASYM

Figure 2.3: Results of memorization of label noise and performance on test data on CIFAR-10 with 80% symmetric label noise (SYM) and 40% asymmetric label noise (ASYM). The memorization is defined by the fraction of wrongly labeled examples whose predictions are equal to their labels.

Lemma 2.2.5. *Let* $R(X) = \inf_g \mathbb{E}_{X,Y}[\mathcal{L}(g(X), Y)]$ *be the minimum risk over the joint distribution* $X \times Y$*, where* $\mathcal{L}(p, y) = \sum_{i=1}^{\mathcal{Y}} y^{(i)} \log p^{(i)}$ *is a CE loss and* g *is a function mapping from input space to label space,* $p^{(i)}$ *with upper script* i *represents the* i*th entry for the probabilistic output of* X*. Let* $R(Z^\star) = \inf_{g'} \mathbb{E}_{Z^\star,Y}[\mathcal{L}(g'(Z^\star), Y)]$ *be the minimum risk over the joint distribution* $Z^\star \times Y$ *and* g' *maps from representation space to label space. Suppose the joint distribution* $D(X, Y, \tilde{Y})$ *is* (ϵ, γ)*-Distribution. Then,*

$$R(Z^\star) \leq R(X) + \epsilon.$$

The proof is provided in 2.B.4. The ϵ comes from the definition of the joint distribution.

To show the robustness and performance of the contrastive (CTR) representation $Z^\star$, we empirically compare it to the representation learned by the CE loss. We first use clean labels to train neural networks with different loss functions. Then we initialize the parameters of final linear classifiers and fine tune them with noisy labels. We denote the memorization by the fraction of corrupted examples whose predictions are equal to their labels. Figure 2.3 illustrates the improved performance and robustness in terms of test accuracy and reduced memorization with the CTR representation.

Remark Figure 2.3 indicates that the memorization subsides down as the training progresses, whereas in previous literature, the memorization increases as training progresses. We explain that, conventionally, the memorization is observed and proved in **over-parameterized** models. Under this setting, the fraction of examples that memorized by the model will increase as training progresses. However, the memorization in this work is measured on a linear classifier on top of frozen data representations, where ratio of number parameters and the sample size is ~ 0.1, which is underparameterized.

2.3 Algorithm

In practice, as we are only given a noisy data set, we do not know if a label is clean or not. Consequently, simply minimizing Eq. (2.1) can lead to deteriorated performances. To see this, note that Eq. (2.1) is activated only when $\mathbb{1}\{y_i = y_j\} = 1$. Thus, two representations from different classes will be pulled together when there are noisy labels.

Since deep networks first fit examples with clean labels and the probabilistic outputs of these examples are higher than examples with corrupted labels [4, 69], one straightforward approach to tackle this issue is to replace the indicator function with a more reliable criterion $\mathbb{1}\{p_i^\top p_j \geq \tau\}$:

$$\mathcal{L}'_{\text{ctr}}(x_i, x_j) = -(\langle \tilde{q}_i, \tilde{z}_j \rangle + \langle \tilde{q}_j, \tilde{z}_i \rangle)\mathbb{1}\{p_i^\top p_j \geq \tau\}, \tag{2.6}$$

where p_i is the probabilistic output produced by linear classifier on the representation of image x_i and τ is a confidence threshold. However, minimizing Eq. (2.6) only helps representation learning during the early stage. After that period, examples with corrupted labels will dominate the learning procedure since the magnitudes of gradient from correct contrastive pairs overwhelm that from wrong contrastive pairs. In particular, given two clean examples x_i, x_j with $y_i = y_j$ and a wrongly labeled example x_m with $\tilde{y}_m = y_i = y_j$, during the early stage, representations $\tilde{q}_i^\top \tilde{q}_j \to 1$ and $\tilde{q}_i^\top \tilde{q}_m \approx 0$. After the early stage, deep networks starts to fit wrongly labeled data. At this moment, the wrong contrastive pairs (x_i, x_m) and (x_j, x_m) are wrongly pulled together and they impair the representation learning instead of the correct pair (x_i, x_j):

$$\left\| \frac{\partial \mathcal{L}'_{\text{ctr}}(x_i, x_m)}{\partial q_i} \right\|_2^2 = c_i(\underbrace{1 - \tilde{q}_i^\top \tilde{q}_m}_{\approx 1}) \gg c_i(\underbrace{1 - \tilde{q}_i^\top \tilde{q}_j}_{\approx 0}) = \left\| \frac{\partial \mathcal{L}'_{\text{ctr}}(x_i, x_j)}{\partial q_i} \right\|_2^2, \tag{2.7}$$

where $c_i = 1/\|q_i\|_2^2$ and we take h as an identity function for simplicity. The proof is shown in supplementary materials.

To address this issue, we propose the following regularization function to avoid the negative effects from wrong contrastive pairs:

$$\widetilde{\mathcal{L}}_{\text{ctr}}(x_i, x_j) = \Big(\log(1 - \langle \tilde{q}_i, \tilde{z}_j \rangle) + \log(1 - \langle \tilde{q}_j, \tilde{z}_i \rangle) \Big) \mathbb{1}\{p_i^\top p_j \geq \tau\}. \tag{2.8}$$

Eq. (2.8) still aims to learn similar representations for data with the same true labels. Since the minimum of Eq. (2.8) is the same as the maximum of Eq. (2.1), our theoretical results about $Z^\star$ still hold. Moreover, the gradient analysis of Eq. (2.8) is given by

$$\left\| \frac{\partial \widetilde{\mathcal{L}}_{\text{ctr}}(x_i, x_j)}{\partial q_i} \right\|_2^2 = c_i(1 + \tilde{q}_i^\top \tilde{q}_j), \tag{2.9}$$

which indicates that the gradient in L2 norm increases if $\tilde{q}_i$ and $\tilde{q}_j$ approach to each other. In other words, the gradient from the correct pair (x_i, x_j) is larger than the gradient from the wrong pair (x_i, x_m) $(1 + \tilde{q}_i^\top \tilde{q}_j > 1 + \tilde{q}_i^\top \tilde{q}_m \approx 1)$ during the learning procedure. Compared to the gradient given by Eq. (2.7), our proposed regularization function does not suffer from the gradient domination by wrong pairs.

Finally, the overall objective function is given by

$$\mathcal{L} = \mathcal{L}_{\text{ce}} + \lambda \widetilde{\mathcal{L}}_{\text{ctr}}, \tag{2.10}$$

where $\widetilde{\mathcal{L}}_{\text{ctr}}$ serves as a contrastive regularization (CTRR) on representations and λ controls the strength of the regularization.

Method	CIFAR-10						
	Sym.						Asym.
	0%	20%	40%	60%	80%	90%	40%
CE	$93.97_{\pm 0.22}$	$88.51_{\pm 0.17}$	$82.73_{\pm 0.16}$	$76.26_{\pm 0.29}$	$59.25_{\pm 1.01}$	$39.43_{\pm 1.17}$	$83.23_{\pm 0.59}$
Forward	$93.47_{\pm 0.19}$	$88.87_{\pm 0.21}$	$83.28_{\pm 0.37}$	$75.15_{\pm 0.73}$	$58.58_{\pm 1.05}$	$38.49_{\pm 1.02}$	$82.93_{\pm 0.74}$
GCE	$92.38_{\pm 0.32}$	$91.22_{\pm 0.25}$	$89.26_{\pm 0.34}$	$85.76_{\pm 0.58}$	$70.57_{\pm 0.83}$	$31.25_{\pm 1.04}$	$82.23_{\pm 0.61}$
Co-teaching	$93.37_{\pm 0.12}$	$92.05_{\pm 0.15}$	$87.73_{\pm 0.17}$	$85.10_{\pm 0.49}$	$44.16_{\pm 0.71}$	$30.39_{\pm 1.08}$	$77.78_{\pm 0.59}$
LIMIT	$93.47_{\pm 0.56}$	$89.63_{\pm 0.42}$	$85.39_{\pm 0.63}$	$78.05_{\pm 0.85}$	$58.71_{\pm 0.83}$	$40.46_{\pm 0.97}$	$83.56_{\pm 0.70}$
SLN	$93.21_{\pm 0.21}$	$88.77_{\pm 0.23}$	$87.03_{\pm 0.70}$	$80.57_{\pm 0.50}$	$63.99_{\pm 0.79}$	$36.64_{\pm 1.77}$	$81.02_{\pm 0.25}$
SL	$94.21_{\pm 0.13}$	$92.45_{\pm 0.08}$	$89.22_{\pm 0.08}$	$84.63_{\pm 0.21}$	$72.59_{\pm 0.23}$	$51.13_{\pm 0.27}$	$83.58_{\pm 0.60}$
APL	$93.97_{\pm 0.25}$	$92.51_{\pm 0.39}$	$89.34_{\pm 0.33}$	$85.01_{\pm 0.17}$	$70.52_{\pm 2.36}$	$49.38_{\pm 2.86}$	$84.06_{\pm 0.20}$
CTRR	$\mathbf{94.29_{\pm 0.21}}$	$\mathbf{93.05_{\pm 0.32}}$	$\mathbf{92.16_{\pm 0.31}}$	$\mathbf{87.34_{\pm 0.84}}$	$\mathbf{83.66_{\pm 0.52}}$	$\mathbf{81.65_{\pm 2.46}}$	$\mathbf{89.00_{\pm 0.56}}$

Table 2.1: The test accuracy on CIFAR-10 with different noise types and noise levels. All method use the same model PreAct ResNet18 [43] and their results are reported over three runs.

2.4 Experiments

Datasets. We evaluate our method on two artificially corrupted datasets CIFAR-10 [31] and CIFAR-100 [31], and two real-world datasets ANIMAL-10N [113] and Clothing1M [141]. CIFAR-10 and CIFAR1-00 contain $50,000$ training images and $10,000$ test images with 10 and 100 classes, respectively. ANIMAL-10N has 10 animal classes and $50,000$ training images with confusing appearances and 5000 test images. Its estimated noise level is around 8%. Clothing1M has a million training images and $10,000$ test images with 14 classes. Its estimated noise level is around 40%.

Noise generation. For CIFAR-10, we consider two different types of synthetic noise with various noise levels. For **symmetric noise**, each label has the same probability of flipping to any other classes, and we randomly choose r training data with their labels to be flipped for $r \in \{20\%, 40\%, 60\%, 80\%, 90\%\}$. For **asymmetric noise**, following [11], we flip labels between TRUCK→AUTOMOBILE, BIRD→AIRPLANE, DEER→HORSE, and CAT↔DOG. we randomly choose 40% training data with their labels to be flipped according to the asymmetric labeling rule. For CIFAR-100, we test our method with symmetric noise with the noise level $r \in \{20\%, 40\%, 60\%, 80\%\}$. From statistical point of view, if a sample is noisily labeled according to symmetric label noise, then its label follows uniform distribution. If a sample is noisily labeled according to asymmetric label noise, then its label follows the distribution where the probability of observed label given ground-truth is either 0 or 1, For example $\Pr[\tilde{Y} = j|Y = i] = c_{ij}$, where c_{ij} is either 0 or 1.

Baseline methods. To evaluate our method, we mainly compare our robust loss function to other robust loss function methods: 1) CE loss. 2) Forward correction [95], which corrects loss values by a estimated noise transition matrix. 3) GCE [161], which takes advantages of both MAE loss and CE loss and designs a robust loss function. 4) Co-teaching [38], which maintains two networks and uses small-loss examples to update. 5) LIMIT [40], which introduces noise to gradients to avoid memorization. 6) SLN [11], which adds Gaussian noise to noisy labels to

Method	CIFAR-100					
	Sym.					Asym.
	0%	20%	40%	60%	80%	40%
CE	$73.21_{\pm 0.14}$	$60.57_{\pm 0.53}$	$52.48_{\pm 0.34}$	$43.20_{\pm 0.21}$	$22.96_{\pm 0.84}$	$44.45_{\pm 0.37}$
Forward	$73.01_{\pm 0.33}$	$58.72_{\pm 0.54}$	$50.10_{\pm 0.84}$	$39.35_{\pm 0.82}$	$17.15_{\pm 1.81}$	-
GCE	$72.27_{\pm 0.27}$	$68.31_{\pm 0.34}$	$62.25_{\pm 0.48}$	$53.86_{\pm 0.95}$	$19.31_{\pm 1.14}$	$46.50_{\pm 0.71}$
Co-teaching	$73.39_{\pm 0.27}$	$65.71_{\pm 0.20}$	$57.64_{\pm 0.71}$	$31.59_{\pm 0.88}$	$15.28_{\pm 1.94}$	-
LIMIT	$65.53_{\pm 0.91}$	$58.02_{\pm 1.93}$	$49.71_{\pm 1.81}$	$37.05_{\pm 1.39}$	$20.01_{\pm 0.11}$	-
SLN	$63.13_{\pm 0.21}$	$55.35_{\pm 1.26}$	$51.39_{\pm 0.48}$	$35.53_{\pm 0.58}$	$11.96_{\pm 2.03}$	-
SL	$72.44_{\pm 0.44}$	$66.46_{\pm 0.26}$	$61.44_{\pm 0.23}$	$54.17_{\pm 1.32}$	$34.22_{\pm 1.06}$	$46.12_{\pm 0.47}$
APL	$73.88_{\pm 0.99}$	$68.09_{\pm 0.15}$	$63.46_{\pm 0.17}$	$53.63_{\pm 0.45}$	$20.00_{\pm 2.02}$	$52.80_{\pm 0.52}$
CTRR	$\mathbf{74.36}_{\pm \mathbf{0.41}}$	$\mathbf{70.09}_{\pm \mathbf{0.45}}$	$\mathbf{65.32}_{\pm \mathbf{0.20}}$	$\mathbf{54.20}_{\pm \mathbf{0.34}}$	$\mathbf{43.69}_{\pm \mathbf{0.28}}$	$\mathbf{54.47}_{\pm \mathbf{0.37}}$

Table 2.2: The test accuracy on CIFAR-100 with different noise levels. All method use the same model PreAct ResNet18 [43] and their results are reported over three runs.

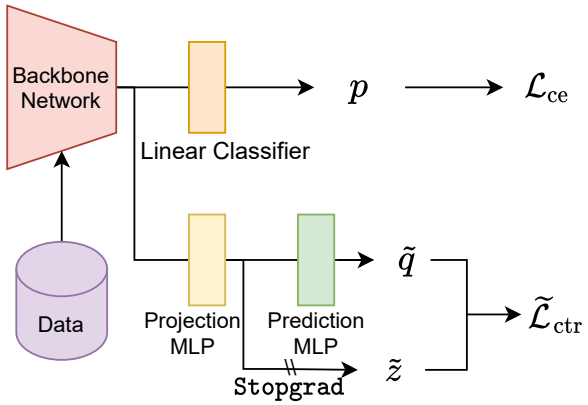


Figure 2.4: Illustration of our framework, where the `Stopgrad` is the stop gradient operator.

combat label noise. 7) SL [134], which uses CE loss and a reverse cross entropy loss (RCE) as a robust loss function. 8) APL (NCE+RCE) [85], which combines two mutually boosted robust loss functions for training.

Implementation. For CIFAR datasets, we use the model PreAct ResNet18. For ANIMAL-10N, we use a random initialized model ResNet18. For Clothing1M, we use an ImageNet pre-trained model ResNet18. The projection MLP is 3-layer MLP and the prediction MLP is 2-layer MLP as proposed in Simsiam [16]. We illustrate our framework in Figure 2.4. We use weak augmentations $\mathcal{A}_w : \mathcal{X} \to \mathcal{X}$ including random resized crop and random horizontal flip for optimizing the cross entropy loss $\mathcal{L}_{\text{ce}}$. Following SimSiam, we use a strong augmentation $\mathcal{A}_s : \mathcal{X} \to \mathcal{X}$ applied on images twice for optimizing the contrastive regularization term $\widetilde{\mathcal{L}}_{\text{ctr}}$. Specifically, $\{z_i\} = f(\mathcal{A}_s(\{x_i\}))$ and $\{q_i\} = h(f(\mathcal{A}_s(\{x_i\})))$ for every example x_i, where one strong augmented image is for calculating z and another is for calculating q. We include more implementation details and algorithms in Appendix 2.A.

Method	ANIMAL-10N	Clothing1M
CE	$83.18_{\pm 0.15}$	$70.88_{\pm 0.45}$
Forward	$83.67_{\pm 0.31}$	$71.23_{\pm 0.39}$
GCE	$84.42_{\pm 0.39}$	$71.34_{\pm 0.12}$
Co-teaching	$85.73_{\pm 0.27}$	$71.68_{\pm 0.21}$
SLN	$83.17_{\pm 0.08}$	$71.17_{\pm 0.12}$
SL	$83.92_{\pm 0.28}$	$72.03_{\pm 0.13}$
APL	$84.25_{\pm 0.11}$	$72.18_{\pm 0.21}$
CTRR	$\mathbf{86.71_{\pm 0.15}}$	$\mathbf{72.71_{\pm 0.19}}$

Table 2.3: Test accuracy on the real-world datasets ANIMAL-10N and Clothing1M. For ANIMAL-10N, all methods use a random initialized ResNet18 and pre-trained ResNet18 for Clothing1M. The results are based on three different runs.

Regularization Functions	CIFAR-10					
	0%	20%	40%	60%	80%	90%
$\mathcal{L}'_{\text{ctr}}$ (2.6)	$93.58_{\pm 0.11}$	$86.05_{\pm 0.33}$	$82.34_{\pm 0.25}$	$74.35_{\pm 0.54}$	$54.83_{\pm 1.00}$	$40.96_{\pm 0.99}$
$\widetilde{\mathcal{L}}_{\text{ctr}}$ (2.8)	$\mathbf{94.29_{\pm 0.21}}$	$\mathbf{93.05_{\pm 0.32}}$	$\mathbf{92.16_{\pm 0.31}}$	$\mathbf{87.34_{\pm 0.84}}$	$\mathbf{83.66_{\pm 0.52}}$	$\mathbf{81.65_{\pm 2.46}}$

Table 2.4: The performance of the model with respect to different regularization functions.

2.4.1 CIFAR Results

Table 2.1 and Table 2.2 show the results on CIFAR-10 and CIFAR-100 with various label noise settings. We use PreAct ResNet18 [43] for all methods and report the test accuracy for them based on three runs. Our method achieves the best performance on all tested noise settings. Especially when noise levels reach to 80% or even 90%, our method significantly outperforms other methods. For example, on CIFAR-10 with $r = 90\%$, CTRR maintains a high accuracy of 81.65% compared with the second best one 49.65%.

2.4.2 ANIMAL-10N & Clothing1M Results

Table 2.3 shows the results on the real-world datasets ANIMAL-10N and Clothing1M. All methods use the same model and the results are reported over three runs. We use a random initialized ResNet18 and an ImageNet pre-trained ResNet18 on ANIMAL-10N and Clothing1M, respectively, and the results are reported over three runs. For Clothing1M, following [66, 11], we randomly sample a balanced subset of 20.48K images from the noisy training data and report performance on 10K test images. Our method is superior to other baselines on the two real-world datasets.

Method	20%	40%	60%	80%	90%
Vanilla-SimSiam	$90.76_{\pm 0.12}$	$88.14_{\pm 0.23}$	$83.35_{\pm 0.39}$	$78.66_{\pm 0.29}$	$75.70_{\pm 1.04}$
CTRR (SimSiam)	$93.05_{\pm 0.32}$	$92.16_{\pm 0.31}$	$87.34_{\pm 0.84}$	$83.66_{\pm 0.52}$	$81.65_{\pm 2.46}$
CTRR (SimCLR)	$92.50_{\pm 0.35}$	$90.12_{\pm 0.43}$	$87.41_{\pm 0.83}$	$84.96_{\pm 0.44}$	$79.57_{\pm 1.32}$
CTRR (BYOL)	$93.31_{\pm 0.16}$	$92.12_{\pm 0.16}$	$88.71_{\pm 0.52}$	$86.99_{\pm 0.59}$	$84.31_{\pm 0.66}$

Table 2.5: Comparison with other contrasitve learning methods.

2.5 Ablation Studies and Discussions

In this section, we first investigate the effects of hyperparameters. Then we evaluate the regularization functions Eq. (2.8) and Eq. (2.6), respectively. Lastly, we present two ways to enhance our method.

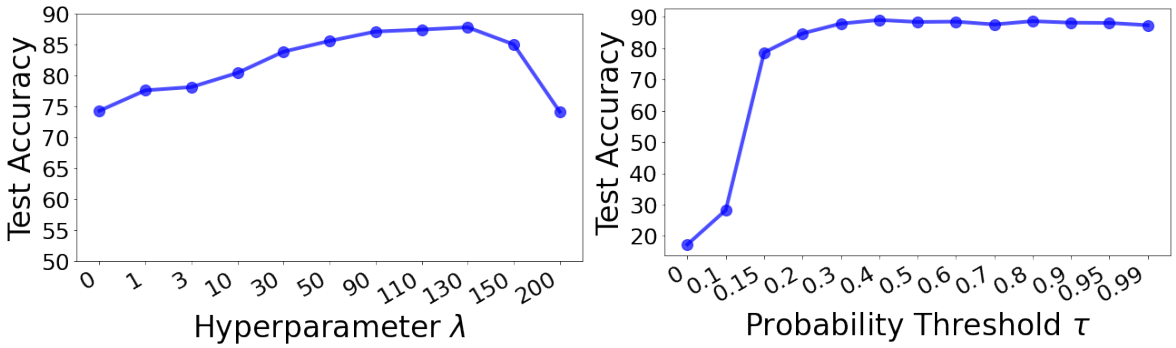


Figure 2.5: Analysis of λ and τ on CIFAR-10 with 60% symmetric label noise

Label Correction Technique	CIFAR-10			
	20%	40%	60%	80%
✗	$93.05_{\pm 0.32}$	$92.16_{\pm 0.31}$	$87.34_{\pm 0.84}$	$83.66_{\pm 0.52}$
✓	$\mathbf{93.32_{\pm 0.11}}$	$\mathbf{92.76_{\pm 0.67}}$	$\mathbf{89.23_{\pm 0.18}}$	$\mathbf{85.40_{\pm 0.93}}$

Table 2.6: ✓/✗ indicates the label correction technique is enabled/disabled.

The effects of λ: The hyperparameter λ controls the strength of the regularization to representations of data. A weak regularization is not able to address the memorization issue, while a strong regularization makes the neural network mainly focus on optimizing the regularization term and ignoring optimizing the linear classifier. Figure 2.5 (left) shows the test accuracy with different λ. The results are in line with the expectation that too strong and too weak regularizations lead to poor performance.

The effects of τ: The τ is the confidence threshold for choosing two examples from the same classes. Many wrong pairs are selected if τ is set too low. Figure 2.5 (right) shows the test accuracy with different τ. When we are too confident about any pairs (τ=0), the model performance is reduced significantly.

The effects of $\widetilde{\mathcal{L}}_{\text{ctr}}$: To study the effect of the proposed regularization function, we compare the performance of Eq. (2.8) to Eq. (2.6). Empirical results are consistent with the previous gra-

Method	CIFAR-10			
	20%	40%	60%	80%
GCE	$91.22_{\pm 0.25}$	$89.26_{\pm 0.34}$	$85.76_{\pm 0.58}$	$70.57_{\pm 0.83}$
CTRR	$93.05_{\pm 0.32}$	$92.16_{\pm 0.31}$	$87.34_{\pm 0.84}$	$83.66_{\pm 0.52}$
CTRR+GCE	$\mathbf{93.94_{\pm 0.09}}$	$\mathbf{93.06_{\pm 0.29}}$	$\mathbf{92.79_{\pm 0.06}}$	$\mathbf{90.25_{\pm 0.40}}$

Table 2.7: The performance of the model with respect to GCE, CTRR and CTRR+GCE.

dient analysis and they are shown in Table 2.4. Our proposed regularization function Eq. (2.8) outperforms Eq. (2.6) by a large margin across all noise levels.

2.5.1 Extending to Other Contrastive Learning Frameworks

We first evaluate the performance of vanilla-SimSiam in the presence of label noise, where we pretrain SimSiam on CIFAR-10 without label information and then finetune the neural network on noisy data. Then we evaluate the performance of CTRR under other contrastive learning frameworks. While CTRR relies on SimSiam framework, the mutual information maximization and the gradient analysis can be easily extended to other contrastive learning frameworks, for example, SimCLR [13] and BYOL [36]. Table 2.5 shows that our method still performs well on other frameworks.

2.5.2 Combination with Other Label Noise Methods

Furthermore, CTRR is orthogonal to label correction techniques [160, 75]. In other words, our method can be integrated with these techniques to further boost learning performances. Specifically, we use the basic label correction strategy following [11] that labels are replaced by weighted averaged of both model predictions and original labels, where weights are scaled sample losses. In Table 2.6, we show that the performance is improved after enabling a simple label correction technique.

Note that GCE [161] is a partial noise-robust loss function implicitly combined with CE and MAE. It is of interest to re-validate the loss function GCE along with our proposed regularization function. We show the performance of a combination of our method and GCE in Table 2.7. With representations induced by our proposed method, there is a significant improvement on GCE, which demonstrates the effectiveness of the learned representations. Meanwhile, the success of this combination implies that our proposed method is beneficial to other partial noise-robust loss functions.

2.6 Conclusion

We present a simple but effective CTRR to address the memorization issue. Our theoretical analysis indicates that CTRR induces noise-robust representations without suffering from the underfitting problem. The empirical results also demonstrate the effectiveness of CTRR. We

have discussed the two possible combinations of existing methods to improve model performance. We believe that CTRR can be jointly used with other existing methods to achieve better performance.

2.A Experiment Details

2.A.1 Algorithm

According to our gradient analysis on two different clean images x_i, x_j with $y_i = y_j$ and a noisy image x_m with $y_m = y_i$, apply the regularization function Eq. (2.8) can avoid representation learning dominated by the wrong contrastive pair (x_i, x_m). The analysis does not cover the same image with two different augmentations. When applying the strong augmentation twice, each image x has two different augmentations x', x''. The contrastive pair (x', x'') will also dominate the representation learning given the property of Eq. (2.8). However, focusing on learning similar representations of (x', x'') does not help to form a cluster structure in representation space. As mentioned in [132], learning this self-supervised representations causes representations of data distributed uniformly on the unit hypersphere. Hence, we want the gradient from the pair (x', x'') to be smaller when their representations approach to each other. We use the original contrastive regularization to regularize the pair (x', x''). The pseudocode of the proposed method is given in Algorithm 1.

2.A.2 Hyperparameters

CIFAR. Our method has two hyperparameters λ and τ. For each noise setting for CIFAR-10, we select the best hyperparameters: λ from {50, 130} and τ from {0.4, 0.8}. For each noise setting for CIFAR-100, we select the best hyperparameters: λ from {50, 90} and τ from {0.05, 0.7}. The batch size is set as 256, and the learning rate is 0.02 using SGD with a momentum of 0.9 and a weight dacay of 0.0005.

ANIMAL-10N & Clothing1M. For ANIMAL-10N, we set $\lambda = 50$, $\tau = 0.8$ and batch size is 256. The learning rate is set as 0.04 with the same SGD optimizer as the CIFAR experiment. For Clothing1M, we set $\lambda = 90$, $\tau = 0.4$ and batch size is 256. The learning rate is set as 0.06 with the same SGD optimizer as above.

2.B Proofs of Theoretical Results

2.B.1 Proof for Theorem 2.2.1

Theorem. *Representations Z learned by minimizing Eq. (2.1) maximizes the mutual information $I(Z; X^+)$.*

Proof. We first decompose the mutual information $I(Z; X^+)$:

$$\begin{aligned} I(Z; X^+) =& \mathbb{E}_{Z,X^+} \log \frac{p(Z|X^+)}{p(Z)} \\ =& \mathbb{E}_{X^+} \mathbb{E}_{Z|X^+} [\log p(Z|X^+)] - \mathbb{E}_{Z,X^+} [p(Z)] \\ =& - \mathbb{E}_{X^+} [H(Z|X^+)] + H(Z). \end{aligned}$$

The first term $\mathbb{E}_{X^+}[H(Z|X^+)]$ measures the uncertainty of $Z|X^+$, which is minimized when Z can be completely determined by X^+. The second term $H(Z)$ measures the uncertainty of Z itself and it is minimized when outcomes of Z are equally likely.

Algorithm 1: CTRR Pseudocode in a PyTorch-like style

```
# Training
# f: backbone + projection mlp
# h: prediction mlp
# g: backbone + softmax linear classifier

for x, y in loader:
    bsz = x.size(0)
    x1, x2 = strong_aug(x), strong_aug(x) # strong random augmentation
    x3 = weak_aug(x) # weak random augmentation
    z1, z2 = f(x1), f(x2)
    q1, q2 = h(z1), h(z2)
    p = g(x3)

    # compute representations
    c1 = torch.matmul(q1, z2.t()) # B X B
    c2 = torch.matmul(q2, z1.t()) # B X B

    # compute contrastive loss for each pair
    m1 = torch.zeros(bsz, bsz).fill_diagonal_(1) # identity matrix
    m2 = torch.ones(bsz, bsz).fill_diagonal_(0) # 1-identity matrix
    # - <i,i> + log(1-<i,j>)
    c1 = -c1*m1 + ((1-c1).log()) * m2
    c2 = -c2*m1 + ((1-c2).log()) * m2
    c = torch.cat([c1, c2], dim=0) # 2B X B

    # compute probability threshold
    probs_thred = torch.matmul(p, p.t()).fill_diagonal_(1).detach() # B X B
    mask = (probs_thred >= tau).float()
    probs_thred = probs_thred * mask
    # normalize the threshold
    weight = probs_thred / probs_thred.sum(1, keepdim=True)
    weight = weight.repeat((2, 1)) # 2B X B

    loss_ctr = (contrast_logits * weight).sum(dim=1).mean(0)
```

We next show that Z can be completely determined by X^+ when minimum of Eq. (2.1) is achieved and uncertainty of Z itself is maintained by an assumption about the framework. By the Cauchy-Schwarz inequality,

$$\begin{aligned}\mathbb{E}_{X,X^+}[\mathcal{L}_{\text{ctr}}(X, X^+)] \geq &\mathbb{E}_{X,X^+}[\, \|\tilde{q}\|_2 \left\|\tilde{z}^+\right\|_2 \\ &+ \left\|\tilde{q}^+\right\|_2 \|\tilde{z}\|_2] = -2.\end{aligned}$$

The equality is attained when $\tilde{q} = \tilde{z}^+$ and $\tilde{q}^+ = \tilde{z}$ for all x, x^+ from the same class. For any three images x_1, x_2, x_3 from the same class, we have:

$$f(x_1) = g(x_3), \quad f(x_2) = g(x_3),$$

where $g = h(f(\cdot))$. We can find $f(x_1) = f(x_2)$ for any images x_1, x_2 from the same class. The result can be easily extended to the general case: $f(X_1) = f(X_2)$ for any $(X_1, Y_1) \sim P(X, Y), (X_2, Y_2) \sim P(X, Y)$ with $Y_1 = Y_2$. Thus Z can be determined by X^+ with the equation $Z = f(X^+)$, which minimizes $\mathbb{E}_{X^+}[H(Z|X^+)]$.

When $p(Z = c_y|Y = y) = \frac{1}{|\mathcal{Y}|}$, the entropy $H(Z)$ is maximized. With extensive empirical results in Simsiam, we assume the collapsed solutions are perfectly avoided by using the SimSiam framework. By this assumption, $c_j \neq c_k$ for any $j \neq k$. The model learns different clusters c_y for different y and representations with different labels have different clusters. Therefore, for a balanced dataset, the outcomes of Z are equally likely and it maximizes the second term $H(Z)$. In summary, the learned representations by Eq. (2.1) maximizes the mutual information $I(Z; X^+)$.

□

2.B.2 Proof for Theorem 2.2.3

Theorem. *Given a distribution $D(X, Y, \tilde{Y})$ that is (ϵ, γ)-Distribution, we have*

$$I(X;Y) - \epsilon \leq I(Z^\star;Y) \leq I(X;Y), \tag{2.11}$$
$$I(Z^\star;\tilde{Y}) \leq I(X;\tilde{Y}) - \gamma + \epsilon. \tag{2.12}$$

Proof. The Theorem builds upon the Theorem 5 from [126]. We first provide the proof for the first inequality, which can also be obtained from [126]. Then we provide the proof for the second inequality.

For the first inequality, by adopting Data Processing Inequality in the Markov Chain $Y \leftrightarrow X \rightarrow Z$, we have $I(X;Y) \geq I(Z;Y)$ for any $Z \in \mathcal{Z}$. Then, we have $I(X;Y) \geq I(Z^\star;Y)$. Since $Z^\star = \arg\max_{Z_\theta} I(Z_\theta;X^+)$, and $I(Z_\theta;X^+)$ is maximized at $I(X;X^+)$, then $I(Z^\star;X^+) = I(X;X^+)$ and $I(Z^\star;X^+|Y) = I(X;X^+|Y)$. Meanwhile, use the result $I(Z^\star;X^+;Y) = I(X;X^+;Y)$, which is given by

$$\begin{aligned} I(Z^\star;X^+;Y) &= I(Z^\star;X^+) - I(Z^\star;X^+|Y) \\ &= I(X;X^+) - I(X;X^+|Y) \\ &= I(X;X^+;Y), \end{aligned}$$

we have

$$\begin{aligned} I(Z^\star;Y) =& I(X;X^+;Y) + I(Z^\star;Y|X^+) \\ =& I(X;Y) - I(X;Y|X^+) + I(Z^\star;Y|X^+). \end{aligned} \tag{2.13}$$

Thus, by Eq. (2.13) and the Definition 2.2.2, we get

$$I(Z^\star;Y) \geq I(X;Y) - I(X;Y|X^+) \geq I(X;Y) - \epsilon \tag{2.14}$$

Now we present the second inequality $I(Z^\star;\tilde{Y}) \leq I(X;\tilde{Y}) - \gamma + \epsilon$.
Similarly, by Eq. (2.13), we have

$$I(Z^\star;\tilde{Y}) = I(X;\tilde{Y}) - I(X;\tilde{Y}|X^+) + I(Z^\star;\tilde{Y}|X^+) \tag{2.15}$$
$$\leq I(X;\tilde{Y}) - \gamma + I(Z^\star;\tilde{Y}|X^+) \tag{2.16}$$
$$\leq I(X;\tilde{Y}) - \gamma + I(Z^\star;Y|X^+) \tag{2.17}$$
$$\leq I(X;\tilde{Y}) - \gamma + \epsilon \tag{2.18}$$

, where the first and the third inequalities are by the definition 2.2.2; the second inequality is by the Data Processing Inequality in the Markov Chain $\tilde{Y} \leftarrow Y \leftrightarrow X \rightarrow Z$.

□

2.B.3 Proof for Lemma 2.2.4

Lemma. *Consider a pair of random variables $(X, \tilde{Y})$. Let $\hat{Y}$ be outputs of any classifier based on inputs Z_θ, and $\tilde{e} = \mathbb{1}\{\hat{Y} \neq \tilde{Y}\}$, where $\mathbb{1}\{A\}$ be the indicator function of event A. Then, we have*

$$\mathbb{E}[\tilde{e}] \geq \frac{H(\tilde{Y}) - I(Z_\theta;\tilde{Y}) - H(\tilde{e})}{\log(|\tilde{\mathcal{Y}}|) - 1}.$$

Proof. If we are given any two of $\{\tilde{e} = 1\}, \hat{Y}, \tilde{Y}$, the other one is known. By the properties of conditional entropy, $H(\tilde{Y}, \tilde{e}|\hat{Y}, Z_\theta)$ can be decomposed into the two equivalent forms.

$$\begin{aligned} H(\tilde{Y}, \tilde{e}|\hat{Y}, Z_\theta) &= H(\tilde{Y}|\tilde{e}, \hat{Y}, Z_\theta) + H(\tilde{e}|\hat{Y}, Z_\theta) \\ &= \underbrace{H(\tilde{Y}|\tilde{e}, \hat{Y}, Z_\theta)}_{0} + H(\tilde{Y}|\hat{Y}, Z_\theta) \end{aligned} \tag{2.19}$$

The first equality can also be decomposed into another form:

$$\begin{aligned} & H(\tilde{Y}, \tilde{e}|\hat{Y}, Z_\theta) \\ =& H(\tilde{Y}|\tilde{e}, \hat{Y}, Z_\theta) + H(\tilde{e}|\hat{Y}, Z_\theta) \\ =& p(\tilde{e} = 1) H(\tilde{Y}|\tilde{e} = 1, \hat{Y}, Z_\theta) \\ & + p(\tilde{e} = 0) \underbrace{H(\tilde{Y}|\tilde{e} = 0, \hat{Y}, Z_\theta)}_{0} + H(\tilde{e}|\hat{Y}, Z_\theta) \\ =& p(\tilde{e} = 1) H(\tilde{Y}|\tilde{e} = 1, \hat{Y}, Z_\theta) + H(\tilde{e}|\hat{Y}, Z_\theta) \end{aligned} \tag{2.20}$$

Relating Eq. (2.19) to Eq. (2.20), we have

$$\begin{aligned} \mathbb{E}[\tilde{e}] &= \frac{H(\tilde{Y}|\hat{Y}, Z_\theta) - H(\tilde{e}|\hat{Y}, Z_\theta)}{H(\tilde{Y}|\tilde{e} = 1, \hat{Y}, Z_\theta)} \\ &\geq \frac{H(\tilde{Y}|\hat{Y}, Z_\theta) - H(\tilde{e}|\hat{Y}, Z_\theta)}{\log(|\mathcal{Y}| - 1)} \\ &\geq \frac{H(\tilde{Y}|\hat{Y}, Z_\theta) - H(\tilde{e})}{\log(|\mathcal{Y}| - 1)} \\ &= \frac{H(\tilde{Y}) - I(\tilde{Y}; Z_\theta, \hat{Y}) - H(\tilde{e})}{\log(|\mathcal{Y}| - 1)} \\ &= \frac{H(\tilde{Y}) - I(\tilde{Y}; Z_\theta) - H(\tilde{e})}{\log(|\mathcal{Y}| - 1)}. \end{aligned}$$

The first inequality is by $H(\tilde{Y}|\tilde{e} = 1, \hat{Y}, Z_\theta) \leq \log(|\mathcal{Y}| - 1)$, where $\tilde{Y}$ can take at most $|\mathcal{Y}| - 1$ values. For the second inequality,

$$\begin{aligned} H(\tilde{e}|\hat{Y}, Z_\theta) &= H(\tilde{e}) - I(\tilde{e}; \hat{Y}, Z_\theta) \\ &\leq H(\tilde{e}). \end{aligned}$$

For the last equality,

$$\begin{aligned} I(\tilde{Y}; Z_\theta, \hat{Y}) =& H(Z_\theta, \hat{Y}) - H(Z_\theta, \hat{Y}|\tilde{Y}) \\ =& H(Z_\theta) + H(\hat{Y}|Z_\theta) \\ & - H(Z_\theta|\tilde{Y}) - H(\hat{Y}|Z_\theta, \tilde{Y}) \\ =& I(Z_\theta, \tilde{Y}) + I(\hat{Y}; \tilde{Y}|Z_\theta) \\ =& I(Z_\theta, \tilde{Y}), \end{aligned}$$

where $I(\hat{Y}; \tilde{Y}|Z_\theta) = 0$ given the Markov Chain $\tilde{Y} \leftarrow Y \leftrightarrow X \rightarrow Z \rightarrow \hat{Y}$:

$$\begin{aligned} I(\hat{Y}; \tilde{Y}|Z_\theta) =& H(\hat{Y}|Z_\theta) - H(\hat{Y}|Z_\theta, \tilde{Y}) \\ =& H(\hat{Y}|Z_\theta) - H(\hat{Y}|Z_\theta) = 0. \end{aligned}$$

□

2.B.4 Proof for Lemma 2.2.5

Lemma. *Let* $R(X) = \inf_g \mathbb{E}_{X,Y}[\mathcal{L}(g(X), Y)]$ *be the minimum risk over the joint distribution* $X \times Y$*, where* $\mathcal{L}(p, y) = \sum_{i=1}^{\mathcal{Y}} y^{(i)} \log p^{(i)}$ *is a CE loss and* g *is a function mapping from input space to label space. Let* $R(Z^\star) = \inf_{g'} \mathbb{E}_{Z^\star,Y}[\mathcal{L}(g'(Z^\star), Y)]$ *be the minimum risk over the joint distribution* $Z^\star \times Y$ *and* g' *maps from representation space to label space. Then,*

$$R(Z^\star) \leq R(X) + \epsilon.$$

Proof. The lemma is given by the variational form of the conditional entropy $H(Y|Z^\star) = \inf_{g'} \mathbb{E}_{Z^\star,Y}[\mathcal{L}(g'(Z^\star), Y)]$ [68, 25]. According to a property of mutual information,

$$I(A; B) = H(A) - H(A|B),$$

we have $R(Z^\star) = H(Y) - I(Z^\star; Y)$. By the results of Theorem 2.2.3,

$$\begin{aligned} R(Z^\star) \leq& H(Y) - I(X; Y) + \epsilon \\ =& H(Y|X) = \inf_g \mathbb{E}_{X,Y}[\mathcal{L}(g(X), Y)]. \end{aligned}$$

□

2.C Gradients of Contrastive regularization Functions

For the contrastive regularization function

$$\mathcal{L}'_{\text{ctr}}(x_i, x_j) = -\left(\frac{q_i}{\|q_i\|_2} \cdot \frac{z_j}{\|z_j\|_2} + \frac{q_j}{\|q_j\|_2} \cdot \frac{z_i}{\|z_i\|_2}\right),$$

we only consider the case $\mathbb{1}\{p_i^\top p_j \geq \tau\} = 1$ because $\mathcal{L}'_{\text{ctr}}(x_i, x_j)$ is not calculated in the algorithm when $\mathbb{1}\{p_i^\top p_j \geq \tau\} = 0$. We assume that h is an identity function and x_i, x_j are from the same class for simplicity.

Let $a = \|q_i\|_2$, $b = q_i$, $x = \frac{z_j}{\|z_j\|_2}$ and $c = \frac{b}{a}$. According to the equation $a^2 = b^\top b$, we differentiate both side of the equation and get

$$2a \, \mathrm{d}a = 2b^\top \, \mathrm{d}b. \tag{2.21}$$

In the meanwhile,

$$\begin{aligned}\partial(\frac{b^\top x}{a}) &= \frac{\mathrm{d}(b^\top x)a - \mathrm{d}ab^\top x}{a^2} \\ &\overset{(2.21)}{=} \frac{ax^\top \mathrm{d}b}{a^2} - \frac{b^\top \mathrm{d}bb^\top x}{a^3} \\ &= \frac{x^\top \mathrm{d}b}{a} - \frac{a^2c^\top xc^\top \mathrm{d}b}{a^3} \\ &= \frac{1}{a}(x^\top - c^\top xc^\top)\,\mathrm{d}b.\end{aligned}$$

Taking a, b, c and x back to the equation, we get the result

$$\frac{\partial \mathcal{L}'_{\mathrm{ctr}}(x_i, x_j)}{\partial q_i} = -\frac{1}{\|q_i\|_2}(\frac{q_j}{\left\|q_j\right\|_2} - (\frac{q_i^\top q_j}{\|q_i\|_2 \left\|q_j\right\|_2})\frac{q_i}{\|q_i\|_2}).$$

Note that $z_i = \texttt{Stopgrad}(q_i)$ because of the identity map h. Let $c_i = 1/\left\|q_i\right\|_2^2$ and then we have

$$\left\|\frac{\partial \mathcal{L}'_{\mathrm{ctr}}(x_i, x_j)}{\partial q_i}\right\|_2^2 = c_i(1 - (\tilde{q}_i^\top \tilde{q}_j)^2).$$

Similarly, for the contrastive regularization function

$$\begin{aligned}\widetilde{\mathcal{L}}_{\mathrm{ctr}}(x_i, x_j) =&\Big(\log(1 - \langle \frac{q_i}{\|q_i\|_2}, \frac{z_j}{\left\|z_j\right\|_2}\rangle) \\ &+ \log(1 - \langle \frac{q_j}{\left\|q_j\right\|_2}, \frac{z_i}{\|z_i\|_2}\rangle)\Big),\end{aligned}$$

$$\begin{aligned}\frac{\partial \widetilde{\mathcal{L}}_{\mathrm{ctr}}(x_i, x_j)}{\partial q_i} =& \frac{1}{1 - \tilde{q}_i^\top \tilde{q}_j}\frac{\partial \mathcal{L}'_{\mathrm{ctr}}(x_i, x_j)}{\partial q_i} \\ =& c_i(1 + \tilde{q}_i^\top \tilde{q}_j).\end{aligned}$$

Chapter 3

How Self-Supervised Learning Helps Learning with Label Noise

3.1 Introduction

Recent self-supervised learning (SSL) with contrastive learning paradigms has achieved great success to learn meaningful data representations without label information [41, 14, 155, 10]. In SSL, any two augmented examples from the *same image* (referred as positive pairs) are mapped to a nearby location in the embedding space, whereas two augmented images from *different images* (referred as negative pairs) are mapped to a distant location. [93, 99, 13]. Empirical evidence demonstrates that representations learned by SSL can be easily adapted to many downstream tasks such as image classification, objection detection, segmentation, and learning with imbalanced datasets [36, 91, 163, 142, 73, 149].

Apart from these applications, in this chapter, we show that learning with label noise can also benefit from representations learned by SSL. The previous methodology focuses on instance-independent label noise, which is unrealistic in practice. For example, blurry images are likely to be mislabeled. We extend the instance-independent to the realistic instance-dependent label noise. We firstly construct a motivating example of instance-dependent label noise, then we prove that a classifier trained on representations learned by SSL with noisy labels is optimal over clean data distribution. Then we systematically analyze the benefits of representations learned by SSL and find two merits of SSL representations: (1) The label noise uniformly spreads over the learned SSL representations. (2) The learned representations exhibit an intrinsic cluster structure that is consistent with true labels.

For point (1), we theoretically show that the label noise is uniformly distributed across the learned representations by SSL in the motivating example, which is easier to address in practice [18, 11, 162]. We further extend the relationship between label noise and the representations learned by SSL to a more general case and provide empirical validation.

As for the point (2), we empirically and theoretically justify that representations learned by SSL exhibit a cluster structure with respect to true labels. Moreover, we show that such the structure encourages the classifier trained on noisy data to be aligned with the optimal classifier obtained from clean distribution.

We further empirically demonstrate that, compared to SSL representations, representations

learned by supervised learning are neither making label noise uniformly distributed across them nor forming a good cluster structure. In particular, representations learned with supervision still depend on the label noise and the they exhibit clusters with respect to noisy labels instead of true labels.

From the algorithmic perspectives, our analysis indicates that we can apply SSL representations as a complementary method to existing label noise methods for learning with label noise. Specifically, we fix representations learned by SSL and then only maintain a linear classifier on the frozen representations by label noise methods. We empirically combine SSL representations with existing label noise methods including a noise-robust loss function method, a sample selection method, and a label correction method, which demonstrates significant improvements.

The main contributions of this chapter are summarized as follows.

- We provide theoretical analysis in a motivating example to show that a classifier trained on representations learned by SSL outperforms a classifier trained by supervised learning in the presence of label noise.
- We systematically analyze why the representations learned by SSL are better to address label noise, where we verify our explanations empirically and theoretically.
- We propose to apply SSL representations as a complementary method to existing label noise methods to handle the label noise, and we conduct extensive experiments to demonstrate the effectiveness of SSL representations.

3.2 Related Work

In this section, we briefly discuss some self-supervised learning work related to this chapter. The literature review of learning with label noise can be found in Chapter 1.

3.2.1 Self-supervised Learning

Representations of images learned by SSL have achieved remarkable success. SimCLR [13] requires a large batch size to contain sufficient in-batch negative pairs and domain-specific augmentations such as Gaussian blur, color distortions, and color jittering. However, a large batch size maybe be infeasible. MoCo [41] solves this issue by introducing a memory bank to store representations of data from previous iterations. BYOL [36] and SimSiam [16] propose new frameworks without using negative pairs so they are able to work with reasonable batch size without using the memory bank. On the other hand, SSL relies on domain-specific image augmentation. That is to assume that image augmentations such as changing colors of images should not affect labels of images in downstream tasks [126]. DACL [129] and I-MIX [64] both leverage MixUp augmentation [157] as domain-agnostic augmentation and they find that SSL methods with both domain-agnostic augmentation and domain-specific augmentations can perform better. A theoretical work on SSL has shown that optimizing a contrastive loss asymptotically optimizes alignment and uniformity properties, where alignment encourages to map

similar images to nearby locations in representation space and uniformity forces representations uniformly distributed on the unit hypersphere [132]. Our work is to study the benefits of SSL representations to learning with label noise.

3.3 A Motivating Example

We first provide a motivating example to show that SSL can be significantly better than supervised learning, which enables us to explore and investigate the benefits of representations learned by SSL.

We first construct a binary classification problem with two linearly separable clusters, where the samples from clusters are artificially flipped according to a label noise function. We denote y_i as the true label for x_i, and assume it is a balanced sample from $\{-1, +1\}$. Then the instance x_i is decided in the following manner:

$$x_i = \begin{cases} e_1\zeta_i + e_2\xi_i, & \text{if } y_i = +1 \\ -e_1\zeta_i - e_2\xi_i, & \text{if } y_i = -1 \end{cases}$$

where $\zeta \sim \mathcal{U}_{[0,4]}$, $\xi \sim \mathcal{U}_{[-1.75,2.25]}$, and $e_1, e_2 \in \mathbb{R}^d$ are two orthogonal unit-norm vectors. We assume $\beta(x, y) = \text{sign}(yx^\top e_2)$ as the instance-dependent label noise function. For each clean example (x_i, y_i), the corresponding noisy example is $(x_i, \tilde{y}_i)$, where $\tilde{y}_i = y_i\beta(x_i, y_i)$. Then we can compute that there are 43.75% mislabeled examples if the noise function $\beta(x, y)$ is adopted.

We assume to use a simple linear classifier parameterized by ω. We use the gradient descent algorithm to learn the parameter ω over the noisy data $\{x_i, \tilde{y}_i\}_{i=1}^n$, with a logistic loss function. Thus, in conventional supervised learning, we have the loss function:

$$\mathcal{L}(\omega) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + \exp\left(-\tilde{y}_i\omega^\top x_i\right)\right).$$

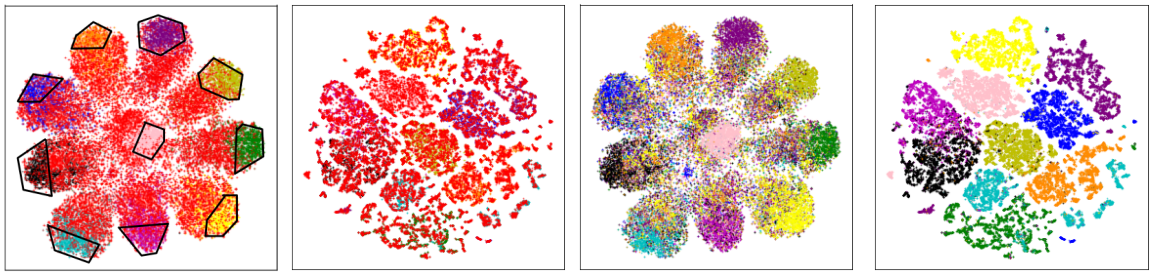
In contrast, in the SSL framework, we first learn a linear representation model with parameter $W \in \mathbb{R}^{1\times d}$ from $\{x_i\}_i^n$ in a self-supervised manner. Specifically, we adopt the linear SSL objective function studied in [73, 39], which tends to pull two positive pairs $(x + \gamma, x + \gamma')$ to nearby locations in the embedding space:

$$W_{\text{SSL}} = \underset{W\in\mathbb{R}^{1\times d}}{\arg\min} -\hat{\mathbb{E}}[(x + \gamma)^\top W^\top W(x + \gamma')] + \frac{1}{2}\left\|W^\top W\right\|_F^2, \tag{3.1}$$

where $\|\cdot\|_F$ is the Frobenius norm, $\hat{\mathbb{E}}$ is an empirical expectation over the data, with γ, γ' are independent and identical $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ random variables. Once the optimal representation model W_{SSL} is obtained, we fix W_{SSL} and then learn a linear classifier parameterized by θ on the top of representations with noisy labels $\{(W_{\text{SSL}}x_i, \tilde{y}_i)\}_i$. Analogous to the supervised learning, we also use the gradient descent with a logistic loss function $\mathcal{L}(\theta)$ to train the classifier.

$$\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + \exp\left(-\tilde{y}_i\theta^\top W_{\text{SSL}}x_i\right)\right).$$

The following theorem states the behavior of linear classifiers on input data $\{(x_i, \tilde{y}_i)\}_i$ and representations of inputs data $\{(W_{\text{SSL}}x_i, \tilde{y}_i)\}_i$, respectively.



(a) SL (noisy labels) (b) SSL (noisy labels) (c) SL (true labels) (d) SSL (true labels)

Figure 3.1: T-SNE of 60% instance-dependent label noise on CIFAR-10. We train a ResNet34 on the noisy data by supervised learning (SL) and we visualize the representations learned by SL in (a) and (c) with respect to noisy labels and true labels, respectively. We also train a SSL representation model ResNet34 without label information and visualize the data representations in (b) and (d) with respect to noisy labels and true labels, respectively. We highlight regions with solid polygons that lightly suffer from the label noise in (a), where red points (label noise) are represented incorrectly labeled examples. In (b), the red points almost uniformly spread over the data representations.

Theorem 3.3.1. *Let $\tilde{\omega}, \tilde{\theta}$ be normalized optimal parameters via gradient descent with logistic loss over the data $\{(x_i, \tilde{y}_i)\}_i$ and $\{(W_{\text{SSL}} x_i, \tilde{y}_i)\}_i$, respectively. Then the generalization accuracy in supervised learning is upper bounded by:*

$$\Pr_{(x,y)}[\text{sign}(\tilde{\omega}^\top x) = y] \leq \frac{9}{16} + \frac{2d}{3n}, \tag{3.2}$$

while the generalization accuracy in SSL is lower bounded by:

$$\Pr_{(x,y)}[\text{sign}(\tilde{\theta}^\top W_{\text{SSL}} x) = y] \geq 1 - 2e^{-n/128}. \tag{3.3}$$

The proof is provided in 3.B.2. Theorem 3.3.1 reveals two interesting facts in the presence of label noise. (1) The prediction accuracy under SSL is guaranteed to be a high value via a provable lower bound. The lower bound could further converge to 1 (perfect prediction without error) when sample size $n \to +\infty$. (2) In contrast, in supervised learning, simply collecting more samples does not guarantee a high accuracy, where the upper bound of the accuracy converges to $9/16$ as $n \to +\infty$.

3.4 Why SSL Works

To show the benefits of SSL in learning with label noise, we start to analyze the learned representation model W_{SSL} from Eq. (3.1).

Proposition 3.4.1. *The optimal solution W_{SSL} in Eq. (3.1) converges in probability to ke_1 with the constant $k > 0$.*

The proof is provided in 3.B.3. The solution W_{SSL} is the span of the vector e_1, which is crucial for learning an optimal classifier in Eq.(3.3). Note that only e_1 determines the true

labels of data x. In fact, the injected label noise depends on non-discriminative feature e_2 but not the discriminative feature e_1. If we orthogonally project data x onto the direction of e_1, the label noise is independent of and is uniformly distributed over the projected data points spanned by e_1. The representation model W_{SSL} exactly maps the data x onto the direction of e_1 orthogonally. Thus, the label noise is uniformly distributed over data representations $W_{\text{SSL}}x$, which makes the label noise easier to address. Specifically:

If the label noise is *uniformly distributed* across the inputs, the classifier trained on data with this label noise can generalize well. Specifically, it can be verified that the optimal Bayes's classifier, $h(x) = \text{sign}(e_1^\top x)$, is also the optimal classifier over the clean distribution. On the other hand, the classifier trained with the supervised learning method from Theorem 3.3.1 is forced to learn spurious correlations between the inputs e_2 and the labels.

Besides, estimating a noise transition matrix is easier when the label noise is uniformly distributed over inputs. In particular, the instance-dependent label noise can be characterized by the noise transition matrix $T(x) \in \mathbb{R}^{|\mathcal{Y}|\times|\mathcal{Y}|}$, where $T(x)_{ij}$ measures the probability of observing a corrupted label j given the true label i and an instance x. The issue of label noise is then solved by estimating the noise transition matrix $T(x)$ [141, 76, 95, 33]. Estimating $T(x)$ for instance-dependent label noise is practically challenging, since $T(x)$ can be different for different x and we may need to parametrize n different $T(x)$ from the noisy dataset of size n by a neural network [18, 139, 8]. In contrast, T is the same for all x for *symmetric label noise* (i.e, label noise is uniformly distributed over data) [95] and we only need to estimate a constant noise transition matrix. Therefore, by estimating a single noise transition matrix instead of parameterizing n noise transition matrices by a neural network, the label noise is easier to solve.

3.4.1 Observations on Real-world Datasets

The benefits of the SSL are not only in the motivating example, but can also be observed in real-world datasets. In this section, we empirically justify the benefits by investigating the CIFAR-10 dataset [56] with 60% instance-dependent label noise. We train a ResNet34 [43] over the noisy data through cross-entropy loss, and visualize the representations in Fig 3.1(a) with respect to noisy labels. The SSL representations with respect to noisy labels are visualized in Fig 3.1(b), where the representation model ResNet34 is trained with self-supervised method MoCov2 [15]. Fig 3.1(a) shows that in the representations learned by supervised learning, the label noise and representations are still dependent (the regions are highlighted by solid polygons), whereas the SSL breaks such the dependency and makes the label noise uniformly distributed across the data representations.

Besides, we also visualize these representations with respect to their true labels in Fig 3.1(c-d). We find that representations learned by SSL exhibit an intrinsic cluster structure that is consistent with the true labels (Fig 3.1(d)). In contrast, Fig 3.1(c) shows that representations learned by supervised learning do not exhibit a cluster structure with respect to true labels. Thus, it further motivates us to explore how the cluster structure learned by SSL helps learning with label noise.

3.5 Cluster Structure and Learning with Label Noise

In this section, we investigate how the cluster structure can help mitigate the label noise. Concretely, we show that for fixed representations, a good cluster structure encourages the classifier to be aligned to the optimal classifier, resulting in better generalization performance.

For simplicity, we use a two-component Gaussian mixture model to describe the clusters of representations, with each cluster representing one class. We assume that representations from class $+1$ are sampled from $\mathcal{N}(\mu, \Sigma)$ and representations from class -1 are sampled from $\mathcal{N}(-\mu, \Sigma)$, where $\mu \in \mathbb{R}^d$ and $\Sigma \in \mathbb{R}^{d\times d}$. In this case, the distance between two clusters is controlled by $\|\mu\|$ and the variance of each cluster is controlled by the sum of eigenvalues of Σ, which is equivalent to the trace of Σ.

We connect the cluster structure to $-\widetilde{\nabla\mathcal{L}}(\omega_0)^\top\tilde{\mu}$, which can characterize the performance of the linear classifier, where $\widetilde{\nabla\mathcal{L}}(\omega_0) = \frac{\nabla\mathcal{L}(\omega_0)}{\|\nabla\mathcal{L}(\omega_0)\|}$, $\tilde{\mu} = \frac{\mu}{\|\mu\|}$, and $\nabla\mathcal{L}(\omega_0)$ is the gradient of the logistic loss computed by the linear classifier (initialized by ω_0). The normalized gradient of the loss $-\widetilde{\nabla\mathcal{L}}(\omega_0)$ represents the direction of steepest descent in the loss function calculated on the noisy data. Given that an optimal classifier obtained from the clean data is $k\mu$ for any scalar $k > 0$, $-\widetilde{\nabla\mathcal{L}}(\omega_0)^\top\tilde{\mu}$ can measure the cosine similarity between the gradient descent direction and the direction where the optimal classifier points. After applying one-step gradient descent to the classifier, the updated classifier is more correlated to the optimal classifier if the cosine similarity is higher. More details can be found in Appendix 3.C.1. This intuitively explains why $-\widetilde{\nabla\mathcal{L}}(\omega_0)^\top\tilde{\mu}$ can be used to measure the performance of the linear classifier. Now we focus on establishing the relationship between $-\widetilde{\nabla\mathcal{L}}(\omega_0)^\top\tilde{\mu}$ and the cluster structure.

As shown in Section 3.4, label noise is uniformly distributed over the data representations. Thus, we define the symmetric label noise function:

$$\beta(x, y) = \begin{cases} -1, & \text{with probability } r \\ +1, & \text{with probability } 1 - r \end{cases}$$

where $0 < r < 1$ controls the noise level. Note that for symmetric label noise, the label noise function $\beta(x, y)$ is independent of the data. The relationship between $-\widetilde{\nabla\mathcal{L}}(\omega_0)^\top\tilde{\mu}$ and the cluster structure is presented in Theorem 3.5.1.

Theorem 3.5.1. *If at least half of examples are clean ($r < \frac{1}{2}$)*

$$-\widetilde{\nabla\mathcal{L}}(\omega_0)^\top\tilde{\mu} \geq \sqrt{\frac{\|\mu\|^2}{c\mathrm{Tr}(\Sigma) + \|\mu\|^2}}(1 - 2r) + o(n^{-1/3}), \tag{3.4}$$

where $\mathrm{Tr}(\Sigma)$ *is the trace of* Σ *and* $c > 0$ *is a constant.*

The proof is provided in 3.C.2. Theorem 3.5.1 provides a lower bound for $-\widetilde{\nabla\mathcal{L}}(\omega_0)^\top\tilde{\mu}$. The larger lower bound means the updated classifier is more correlated to the optimal one. The lower bound can be affected by the noise level r and the following two cluster properties: **1)** the distance between two clusters $\|\mu\|$, **2)** the variance of each cluster $\mathrm{Tr}(\Sigma)$. Without considering any label correction techniques, the noise level r is fixed given a dataset. Therefore, by learning clusters of data representations that are distant to each other (larger $\|\mu\|$) and/or by learning tight representation clusters (smaller $\mathrm{Tr}(\Sigma)$), the classifier generalizes better.

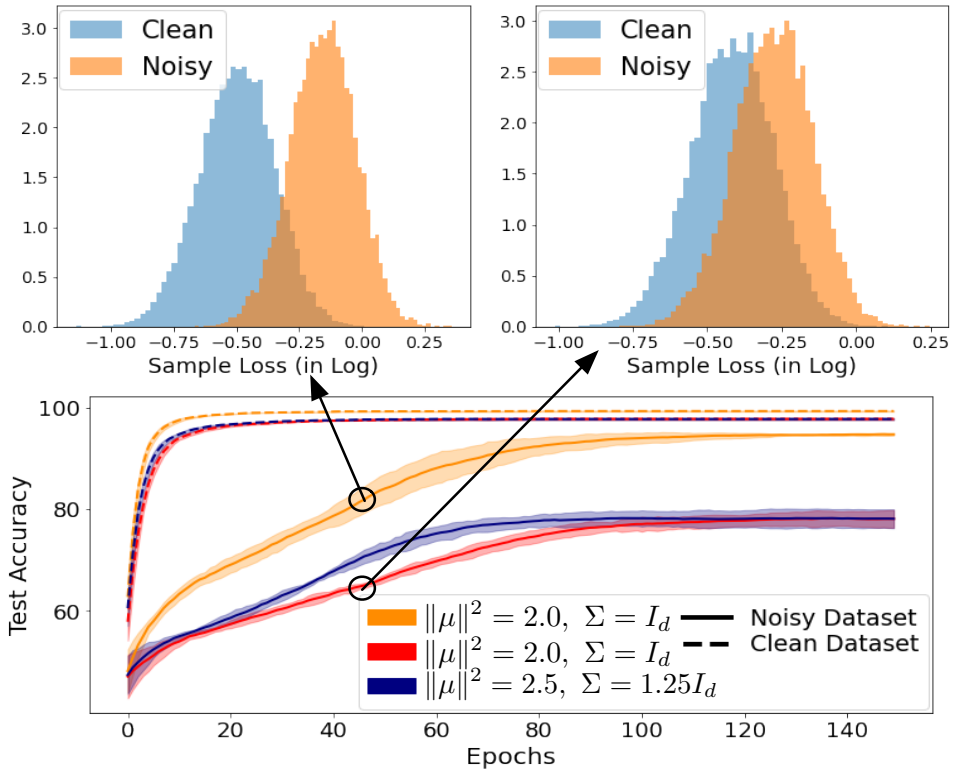


Figure 3.2: Results of linear classifiers trained on synthetic datasets with 40% noise level. We use the dash line to represent the performance of classifiers trained without suffering from label noise and the solid line to represent that with label noise. The histograms are samples loss values at epoch = 50 with respect to whether they are mislabeled.

Remark We remark that the spirits of encouraging a good cluster structure are the same for other forms of label noise such as asymmetric label noise, though their expressions of $-\widetilde{\nabla\mathcal{L}}(\omega_0)^\top\tilde{\mu}$ are different. More details can be found in Appendix 3.C.2.

We empirically justify that linear classifiers achieve better performance when $-\widetilde{\nabla\mathcal{L}}(\omega_0)^\top\tilde{\mu}$ becomes larger. Fig 3.2 shows performances of classifiers trained on data points with different cluster structures given a fixed noise level 40%. Specifically, with the same variance, the classifier trained on clusters with larger distance performs better (orange line v.s. red line). While with the same distance, the classifier trained on tight clusters performs better (orange line v.s. blue line). The two histograms demonstrate that the linear classifier with larger $-\widetilde{\nabla\mathcal{L}}(\omega_0)^\top\tilde{\mu}$ (orange) fits clean examples better, compared with the linear classifier (red). It also highlights that by learning representations with better cluster structure, the classifier generalizes better on clean data distribution.

3.6 Learning Cluster Structure of Representations by SSL

In this section, we rigorously justify that the cluster properties characterized in Theorem 3.5.1 can be achieved by SSL. In particular, we focus on the SSL objective function Eq. (3.5) that has been studied in [132]. Notably, the loss and its variants have been widely adopted in the

SSL such as [13, 41, 15, 93, 39, 125, 94].

$$\mathcal{L}_{\text{ctr}}(f) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{\mathbb{E}_{\substack{u_i\sim P(u|x_i)\\ u_i^+\sim P(u|x_i)}}\left[\left\|f(u_i) - f(u_i^+)\right\|^2\right]}_{\mathcal{L}_{\text{Align}}(i)} \tag{3.5}$$
$$+ \lambda \log\left[\frac{1}{n(n-1)}\sum_{i\neq j} \underbrace{\mathbb{E}_{\substack{u_i\sim P(u|x_i)\\ u_j\sim P(u|x_j)}}\left[e^{-\left\|f(u_i)-f(u_j)\right\|^2}\right]}_{\mathcal{L}_{\text{Uniform}}(i,j)}\right],$$

where λ is a hyper-parameter. $\{x_1, x_2, \cdots, x_n\}$ are input instances, $P(u|x)$ denotes the conditional distribution of an augmented instance u gievn x, and $f(\cdot)$ is a representation network that takes u as an input. Intuitively, Eq. (3.5) aims to minimize the distance between two representations of different views from the same instance, and the $\mathcal{L}_{\text{Uniform}}$ makes representations uniformly distributed over the embedding space [132].

To characterize the cluster properties studied in Theorem 3.5.1, we introduce the notion of δ-cluster closeness.

Definition 3.6.1 (δ-cluster closeness)**.** *Let S_i be the support where $P(u|x_i) > 0$ for any $u \in S_i$. S_i and S_j are δ-cluster close if $P[u \in S_i \cap S_j | x_i] \geq \delta$ for any two different instances x_i, x_j with $y_i = y_j$.*

The notion of δ-cluster closeness is similar to the cluster assumption in [61, 105, 112]. Definition 3.6.1 reveals that the instance augmentations should be rich enough so that any two different augmentation distributions from the same class can be overlapped. Following [129, 64], we apply mixup data augmentation on top of the conventional SSL data augmentations [13] in order to have richer instance augmentations, making Definition 3.6.1 hold with a large δ.

With the notion of δ-cluster closeness, we investigate the relationship between Eq. (3.5) and cluster properties of representations: the distances between any two clusters and the variance of each cluster.

To analyze the cluster properties, we decompose $\mathcal{L}_{\text{ctr}}(f)$ into three different components and we study the effects of them individually.

$$\mathcal{L}_{\text{ctr}}(f) = \frac{1}{n}\sum_{m\in\mathcal{Y}}\sum_{i\in J_m}\mathcal{L}_{\text{Align}}(i)$$
$$+ \lambda \log\left[\frac{1}{n(n-1)}\sum_{\substack{m\in\mathcal{Y}, n\in\mathcal{Y}\\ m\neq n}}\sum_{\substack{i\in J_m\\ j\in J_n}}\mathcal{L}_{\text{Uniform}}(i,j) + \frac{1}{n(n-1)}\sum_{m\in\mathcal{Y}}\sum_{\substack{i,j\in J_m\\ i\neq j}}\mathcal{L}_{\text{Uniform}}(i,j)\right], \tag{3.6}$$

The following lemma indicates that a large distance between any two clusters can be achieved by optimizing $\mathcal{L}_{\text{Uniform}}(i,j)$ for some i, j. We denote J_y by the index set corresponding to true class y.

Lemma 3.6.2. *Let* $\hat{\mu}_i = \sum_{k\in J_i} \frac{f(u_k)}{|J_i|}$, $\hat{\mu}_j = \sum_{k\in J_j} \frac{f(u_k)}{|J_j|}$ *be sample means of cluster* i *and cluster* j *with* $i \neq j$. *Without loss of generality, we assume* $|J_i| = |J_j|$. *Then*

$$\mathbb{E}[\left\|\hat{\mu}_i - \hat{\mu}_j\right\|_2] \geq -\frac{1}{|J_i|}\sum_{k\in J_i} \log\left(\mathcal{L}_{Uniform}(k, g(k))\right)$$

where the function $g : J_i \to J_j$ *is any bijective function, and the expectation is over the data augmentation.*

The proof is provided in 3.D.1. Lemma 3.6.2 indicates that the distance between the cluster i and the cluster j can be lower bounded by $-\log\left(\mathcal{L}_{\text{Uniform}}(k, g(k))\right)$ for $k \in J_i$. Since $-\mathcal{L}_{\text{Uniform}}(k, g(k))$ measures the distance between $f(u_k)$ from cluster i and $f(u_{g(k)})$ from cluster j, minimizing $\mathcal{L}_{\text{Uniform}}(k, g(k))$ for all $k \in J_i$ increases the distance between the cluster i and the cluster j.

On the other hand, the objective function Eq. (3.5) also controls the variance of each cluster. The following lemma helps us understand how the SSL objective function Eq. (3.5) controls variance of each cluster.

Lemma 3.6.3. *Let* $\widehat{\Sigma}_y = \frac{1}{|J_y|}\sum_{i\in J_y}(f(u_i) - \hat{\mu}_i)(f(u_i) - \hat{\mu}_i)^\top$ *be the sample covariance matrix. Suppose Definition 3.6.1 holds. Then for any fixed* $\delta \in (0, 1)$, *we have*

$$\text{Tr}(\mathbb{E}[\widehat{\Sigma}_y]) \leq \frac{2}{\delta|J_y|}\sum_{i\in J_y} \mathcal{L}_{Align}(i),$$

where the expectation is over the data augmentation.

The proof is provided in 3.D.2. Lemma 3.6.3 shows that the variance of cluster y is upper bounded by the term $\sum_{i\in J_y} \mathcal{L}_{\text{Align}}(i)$, where the variance is measured by the sum of eigenvalues for the sample covariance matrix computed by representations from the cluster y. In other words, a small variance of cluster y can be achieved by minimizing $\sum_{i\in J_y} \mathcal{L}_{\text{Align}}(i)$.

Lemma 3.6.2 and Lemma 3.6.3 have shown the effects of the first two components in Eq. (3.6). The last component in Eq. (3.6) serves as a contradiction against the first component. We note that minimizing $\mathcal{L}_{\text{Uniform}}(i, j)$ for i, j from the same cluster undesirably increases the variance of that cluster. This intuition is justified by the following proposition.

Proposition 3.6.4. *Suppose Definition 3.6.1 holds with a fixed* δ. *Then*

$$\log[\frac{1}{n(n-1)}\sum_{m\in\mathcal{Y}}\sum_{\substack{i,j\in J_m \\ i\neq j}} \mathcal{L}_{Uniform}(i, j)] \geq -\alpha\sum_{m\in\mathcal{Y}}\sum_{i\in J_m} \mathcal{L}_{Align}(i),$$

where $\alpha = \frac{2(n/|\mathcal{Y}|-1)}{\delta n(n-1)} > 0$.

The proof is provided in 3.D.3. Proposition 3.6.4 indicates that minimizing the third term in Eq. (3.6) forces the first term in to be larger, which makes the variance of clusters to be larger,

where the strength is controlled by a factor α. We note that the third term is due to $\mathcal{L}_{\text{Uniform}}$ of the instances from the same class and it cannot be eliminated since the label information is not leveraged. The constraint strength is mitigated when α decreases. It is small when instance augmentations are rich enough (δ is large), which also highlights the importance of data augmentations in learning SSL representations.

3.7 Experiment

In this section, we apply SSL representations as a complementary method to existing label noise methods to handle the label noise. We empirically demonstrate the effectiveness of representations learned by SSL. We consider two different self-supervised contrastive learning frameworks: MoCov2 [15] and BYOL [36] to show whether results are sensitive to frameworks. Compared to MoCov2, BYOL does not explicitly compute $\mathcal{L}_{\text{Uniform}}$ but may implicitly compute these terms by designing an exponential moving average of the representation network. Details for the experimental settings and more experiments can be found in Appendix 3.A.

Datasets. We validate our method on two artificially corrupted datasets CIFAR-10 and CIFAR-100 [56] with different types of label noise: symmetric (SYM), asymmetric (ASYM), instance-dependent label noise (IDN) More details about the noise generation approach can be found in Appendix 3.A. We also validate our method on a more realistic real-world dataset ANIMAL-10N [113] which consists of $50,000$ training images with confusing appearances from 10 different classes.

Baseline. We evaluate the frozen representations by linear evaluation. In particular, we maintain the linear classifier on the frozen representations with three different types algorithms: robust loss function GCE [161], sample selection Co-teaching [38] and label correction ELR [75]. GCE uses a loss function that aims to address the memorization issue for incorrectly labeled examples. Co-teaching selects clean examples to update the neural network based on the small-loss criterion. ELR introduces a regularization term for pseudo labels and model predictions.

3.7.1 Main Results

Table 3.1 and Table 3.2 show the results of instance-independent label noise on CIFAR-10 and CIFAR-100, respectively. While Table 3.3 and Table 3.4 show the results of instance-dependent label noise on CIFAR-10 and CIFAR-100. All experiments on CIFAR datasets are conducted by the neural network ResNet34. Table 3.5 shows the results on ANIMAL-10N with the neural network ResNet50. Both MoCov2 and BYOL SSL representations can improve efficacy to a wide range of label noise methods: robust loss function methods, sample selection methods and label correction methods. Results demonstrate that training a linear classifier on frozen SSL representations over noisy datasets is significantly better for addressing label noise than training a whole neural network over noisy datasets.

Cluster Structure. We evaluate our cluster structure of SSL representations learned by MoCov2 on CIFAR-10. Learning SSL representations do not leverage the label information, so the representations are invariant to label noise, whereas representations learned by supervised

Table 3.1: Test accuracy on CIFAR-10 and CIFAR-100 datasets with SYM label noise over different noise levels.

Dataset	CIFAR-10					CIFAR-100				
Noise Level	20%	40%	60%	80%	90%	20%	40%	60%	80%	90%
GCE	93.16±0.18	90.11±0.27	82.35±0.29	74.95±0.51	54.34±0.81	71.71±0.09	67.72±0.19	59.5±0.43	35.8±0.62	14.04±0.97
MoCo-GCE	**95.74**±0.07	**95.67**±0.06	**95.58**±0.04	**95.36**±0.08	**94.68**±0.24	**75.21**±0.03	**74.89**±0.08	**73.36**±0.09	**71.91**±0.31	**68.22**±0.72
BYOL-GCE	**95.55**±0.02	**95.46**±0.05	**95.32**±0.06	**95.11**±0.08	**94.66**±0.16	**73.53**±0.03	**72.04**±0.04	**71.43**±0.09	**69.40**±0.20	**65.94**±0.26
CT	93.66±0.17	92.22±0.16	70.51±0.22	39.75±0.88	27.34±0.98	72.69±0.14	68.81±0.19	61.15±0.28	16.40±0.44	8.22±1.46
MoCo-CT	**95.43**±0.07	**95.37**±0.08	**95.19**±0.23	**91.97**±0.80	**87.65**±1.65	**73.86**±0.07	**73.37**±0.12	**72.59**±0.41	**67.79**±0.92	**62.69**±2.18
BYOL-CT	**95.13**±0.02	**94.93**±0.04	**94.71**±0.03	**93.58**±0.55	**87.35**±1.37	**72.19**±0.05	**71.33**±0.18	**69.49**±0.08	**55.55**±3.28	**52.65**±1.22
ELR	93.53±0.10	93.11±0.14	92.22±0.16	85.74±0.52	54.27±1.06	69.64±0.39	65.16±0.30	60.88±0.32	24.92±0.52	10.22±0.76
MoCo-ELR	**95.88**±0.05	**95.81**±0.04	**95.74**±0.03	**95.65**±0.02	**95.60**±0.09	**72.89**±0.39	**72.74**±0.06	**71.74**±0.11	**70.47**±0.19	**66.75**±0.22
BYOL-ELR	**95.55**±0.02	**95.43**±0.03	**95.30**±0.06	**95.11**±0.05	**95.12**±0.10	**72.48**±0.03	**71.73**±0.06	**70.35**±0.10	**68.45**±0.10	**63.70**±0.14

Table 3.2: Test accuracy on CIFAR-10 and CIFAR-100 datasets with ASYM label noise over different noise levels.

Dataset	CIFAR-10					CIFAR-100				
Noise Level	10%	20%	30%	40%	45%	10%	20	30%	40%	45%
GCE	93.0±0.10	91.92±0.23	90.85±0.28	89.44±0.44	85.51±0.59	73.52±0.08	70.05±0.31	65.8±0.35	53.49±0.53	44.08±1.22
MoCo-GCE	**95.63**±0.04	**95.37**±0.08	**95.00**±0.31	**93.30**±0.26	**88.31**±0.41	**74.54**±0.04	**73.60**±0.12	**72.63**±0.13	**66.27**±0.24	**56.12**±0.68
BYOL-GCE	**95.50**±0.02	**95.23**±0.14	**94.83**±0.19	**93.56**±0.36	**90.69**±0.16	**73.17**±0.04	**72.12**±0.09	**70.75**±0.14	**65.09**±0.13	**53.96**±0.50
CT	94.40±0.03	93.32±0.11	90.27±0.15	69.47±0.21	66.08±0.32	73.88±0.04	69.88±0.21	64.64±0.68	55.22±0.71	48.22±1.00
MoCo-CT	**95.37**±0.07	**95.25**±0.09	**94.33**±0.16	**92.29**±0.32	**86.79**±0.52	**73.48**±0.11	**72.02**±0.26	**69.36**±0.41	**63.30**±0.73	**55.70**±1.58
BYOL-CT	**95.48**±0.03	**94.14**±0.72	**94.04**±0.24	**90.72**±0.72	**87.33**±1.23	72.01±0.08	**70.46**±0.04	**66.22**±0.37	54.97±0.94	**46.62**±1.24
ELR	93.90±0.08	93.26±0.10	92.52±0.13	90.93±0.16	**88.49**±0.24	73.89±0.07	73.44±0.20	72.90±0.19	70.62±0.34	65.62±1.31
MoCo-ELR	**95.73**±0.02	**95.69**±0.04	**94.83**±0.12	82.62±1.15	78.92±0.95	**74.87**±0.05	**74.51**±0.10	**73.75**±0.08	**72.26**±0.05	**67.11**±0.28
BYOL-ELR	**95.59**±0.03	**95.49**±0.07	**95.40**±0.03	**94.72**±0.04	86.91±1.73	73.44±0.05	72.95±0.04	71.81±0.05	69.18±0.08	63.27±0.12

learning (SL) are sensitive to label noise. We compare the cluster structure of SSL to that of SL in different label noise settings in Figure 3.3. We find that SSL representations (red) have a better cluster structure than SL representations obtained with different label noise. We note that although the distances between clusters of SL representations (green) learned with IDN are slightly larger than that of SSL representations, the variance of each cluster is 10 times larger. We also highlight the baseline cluster structure with purple, which is trained by SL method without label noise.

Fine-tuned Performance. Following [13], we fine-tune the MoCov2 representation network on CIFAR-10 by GCE algorithm. For noise-free classification tasks, fine-tuning usually outperforms linear evaluation on various classification datasets [13, 36]. However, Figure 3.4 illustrates that linear evaluation (orange) performs better than fine-tuning (blue) in the presence of label noise. When there exists the label noise and the representations are not frozen, fine-tuning degrades the performance of the neural network. We hypothesize that fine-tuning the learned SSL representations destroys the cluster structure with respect to true labels, leading to poor performance.

The effects of MixUp augmentation. We study the importance of mixup data augmentations. Our analysis indicates the importance of keeping larger δ in Lemma 3.6.3 and Proposition 3.6.4 by data augmentation. Without MixUp, the Definition 3.6.1 holds with smaller δ. With MixUp enabled, the bound in Lemma 3.6.3 is tighter and the negative effects in Propo-

Table 3.3: Test accuracy on CIFAR-10 and CIFAR-100 datasets with IDN label noise over different noise levels.

Dataset	CIFAR-10					CIFAR-100				
Noise Level	20%	40%	60%	80%	90%	20%	40%	60%	80%	90%
GCE	90.05±0.29	80.35±0.34	66.94±0.51	49.38±0.66	34.49±0.97	69.58±0.16	60.48±0.32	44.63±0.62	28.34±0.84	14.18±1.29
MoCo-GCE	**95.41**±0.07	**95.05**±0.10	**94.35**±0.21	**91.87**±0.33	**87.72**±0.28	**74.19**±0.04	**72.48**±0.09	**70.47**±0.14	**66.67**±0.46	**61.67**±0.48
BYOL-GCE	**95.22**±0.08	**94.80**±0.09	**94.26**±0.25	**92.88**±0.22	**90.33**±0.77	**72.69**±0.06	**70.86**±0.10	**68.24**±0.10	**65.20**±0.20	**60.06**±0.41
CT	91.50±0.35	85.95±0.38	74.09±0.52	30.79±0.82	22.35±0.92	69.81±0.18	62.59±0.31	52.11±0.65	16.10±0.70	7.91±0.57
MoCo-CT	**95.23**±0.37	**94.68**±0.31	**94.07**±0.58	**83.91**±0.62	**77.87**±2.69	**73.39**±0.22	**72.15**±0.27	**70.14**±0.54	**66.26**±1.08	**58.34**±0.68
BYOL-CT	**94.97**±0.07	**94.52**±0.18	**94.11**±0.14	**89.09**±0.08	**71.72**±1.47	**71.59**±0.04	**70.15**±0.15	**66.73**±0.72	**57.79**±0.45	**49.88**±1.23
ELR	93.54±0.04	93.20±0.18	92.07±0.20	73.27±0.55	41.39±0.80	70.11±0.32	67.16±0.70	58.11±0.67	21.96±0.74	10.28±1.07
MoCo-ELR	**95.77**±0.07	**95.70**±0.10	**95.65**±0.07	**95.58**±0.04	**91.35**±1.91	**72.74**±0.04	**71.56**±0.12	**69.69**±0.22	**65.94**±0.59	**59.80**±0.84
BYOL-ELR	**95.45**±0.02	**95.25**±0.03	**95.08**±0.04	**95.07**±0.06	**94.91**±0.10	**72.11**±0.18	**70.64**±0.32	**68.72**±0.24	**63.75**±0.50	**57.54**±0.48

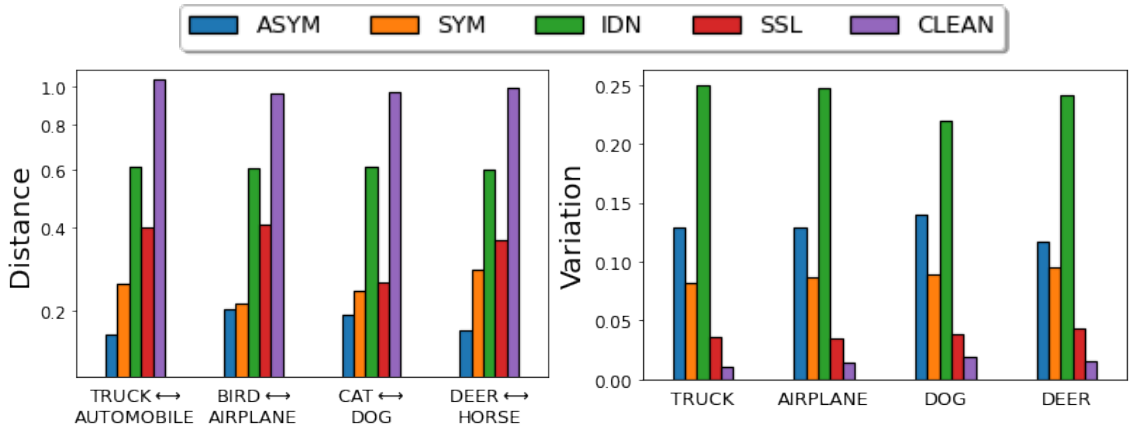
Table 3.4: Test accuracy on CIFAR-10 and CIFAR-100 datasets with IDN-ASYM label noise over different noise levels.

Dataset	CIFAR-10					CIFAR-100				
Noise Level	10%	20%	30%	40%	45%	10%	20	30%	40%	45%
GCE	87.02±0.16	77.40±0.29	68.63±0.68	57.85±0.81	54.01±0.92	71.01±0.12	62.42±0.18	52.48±0.33	44.69±0.78	40.02±0.65
MoCo-GCE	**95.20**±0.02	**94.89**±0.05	**93.28**±0.21	**84.26**±0.43	**71.66**±1.10	**73.90**±0.04	**71.52**±0.57	**69.12**±0.21	**61.69**±0.87	**52.17**±2.23
BYOL-GCE	**95.19**±0.05	**94.85**±0.10	**93.66**±0.21	**85.57**±0.37	**70.44**±1.44	**71.58**±0.14	**69.61**±0.13	**67.22**±0.08	**59.33**±0.77	**50.37**±1.57
CT	87.75±0.28	78.37±0.30	69.31±0.55	60.48±0.57	54.62±0.80	71.97±0.09	64.33±0.13	55.61±0.26	47.12±0.32	42.23±0.46
MoCo-CT	**94.73**±0.45	**93.25**±0.42	**88.92**±0.64	**74.67**±0.91	**61.13**±1.37	**72.77**±0.32	**69.61**±0.49	**66.52**±0.80	**59.37**±1.05	**50.43**±1.43
BYOL-CT	**94.88**±0.06	**93.55**±0.07	**90.99**±0.49	**74.80**±0.39	**63.67**±1.07	**69.54**±0.03	**67.69**±0.12	**64.44**±0.19	**59.23**±0.32	**51.65**±0.94
ELR	94.08±0.05	93.97±0.08	93.91±0.14	93.79±0.20	77.76±2.24	72.21±0.23	71.96±0.24	71.83±0.41	**70.96**±0.43	**67.33**±0.45
MoCo-ELR	**95.72**±0.05	**95.60**±0.04	**95.46**±0.03	**95.23**±0.09	**95.04**±0.20	**73.90**±0.68	**73.16**±0.57	**72.69**±0.35	70.11±0.64	64.51±1.09
BYOL-ELR	**95.39**±0.02	**95.30**±0.03	**95.20**±0.07	**94.99**±0.14	**85.19**±0.32	70.58±0.12	69.80±0.09	68.60±0.18	66.95±0.35	63.32±0.61

sition 3.6.4 is mitigated. Results in Figure 3.5 indicates that applying MixUp augmentation significantly improves the performance on noisy datasets.

3.8 Conclusion

We provide a simple but effective method to address label noise. We first construct a motivating example to theoretically show that the classifier learned on SSL representations can generalize well. By further investigating the SSL representations learned for label noise, we find that: (1) the label noise is uniformly distributed over the data representations. (2) Representations learned by SSL exhibits good cluster properties, which encourages the linear classifier to be aligned with the optimal classifier. From algorithmic perspectives, we demonstrate that SSL representations can be applied as a complementary method to existing label noise methods with extensive experiments.

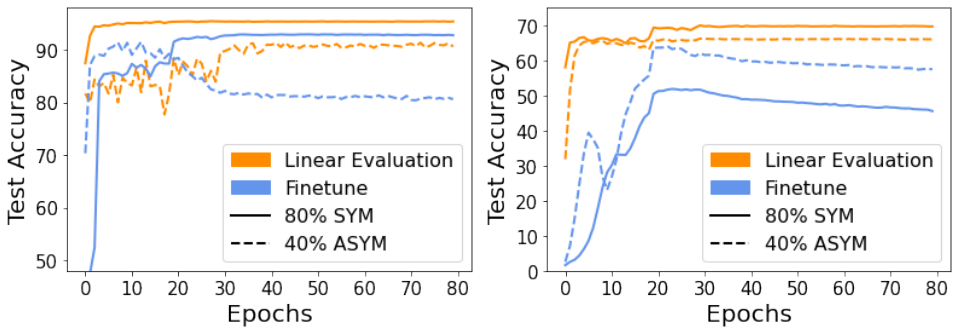


(a) Distances between different clusters (b) Variation of clusters

Figure 3.3: Illustration of cluster structures for CIFAR-10 dataset. Representations are learned in different label noise settings: ASYM (blue) means 40% asymmetric label noise; SYM (orange) means 60% symmetric label noise; IDN (green) means 60% instance-dependent label noise, and we visualize distances between two clusters in (a) and variance of each cluster in (b). The cluster structure (purple) serves as a baseline that representations are trained by supervised learning without label noise.

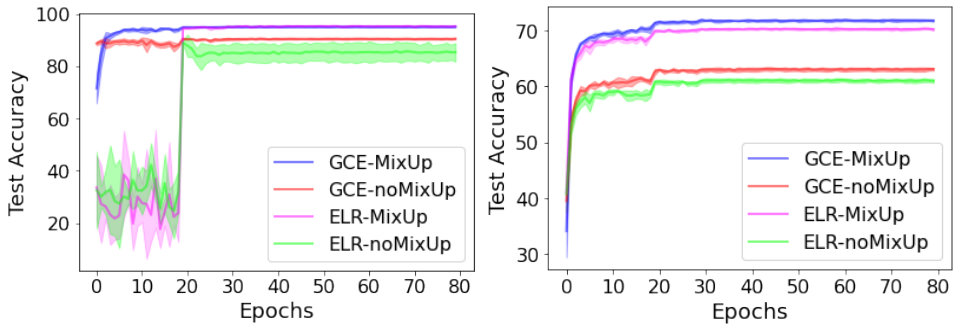
Table 3.5: Test accuracy on ANIMAL-10N

Method	ANIMAL-10N	Method	ANIMAL-10N	Method	ANIMAL-10N
GCE	84.58±0.27	CT	86.93±0.29	ELR	86.52±0.32
MoCo-GCE	**87.35**±0.12	MoCo-CT	**87.66**±0.14	MoCo-ELR	**88.51**±0.12
BYOL-GCE	**88.42**±0.10	BYOL-CT	**88.36**±0.08	BYOL-ELR	**88.68**±0.15



(a) CIFAR-10 (b) CIFAR-100

Figure 3.4: Comparison of linear evaluation and fine-tuning with GCE algorithm.



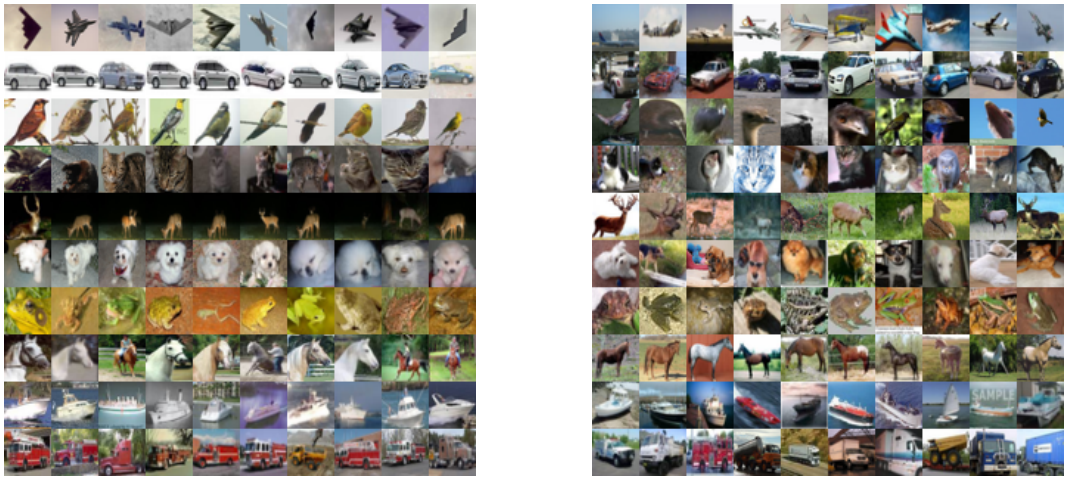
(a) CIFAR-10 (b) CIFAR-100

Figure 3.5: Comparison of SSL methods with and without MixUp component enabled on 80% symmetric label noise.

3.A Experiment Details

3.A.1 Noise Generation

For symmetric label noise, we randomly select a proportion of examples and then flip their labels to all possible labels with equal probabilities. Following [11], for asymmetric label noise in CIFAR-10, we randomly select a proportion of examples and flip their labels between TRUCK→AUTOMOBILE, BIRD→AIRPLANE, DEER→HORSE, and CAT↔DOG. For asymmetric label noise in CIFAR-100, we also randomly select a proportion of examples and but flip their labels into the next class circularly. For instance-dependent label noise, we follow the intuition that mislabeled images share visually similar patterns [141]. To this end, we randomly choose an anchor image for each class, then we choose some similar images to the anchor image and flip their labels like symmetric label noise, where the similarity is measure by L2 norm. To demonstrate this instance-dependent label noise is reasonable, we visualize a part of images from CIFAR-10 with similar visual patterns in Figure 3.6.



(a) Visually similar images (b) Randomly selected images

Figure 3.6: We artificially corrupted images with similar visual patterns. More specifically, we first randomly choose an anchor point for each class, shown in the first column of (a), then we corrupted images that are similar to these anchor images. In contrast, (b) shows the randomly selected images for each class, which does not share similar patterns.

For IDN-SYM, we randomly assign labels to these visually similar images. Asymmetric label noise is to flip labels between semantically-similar classes. For example, cats are inherently more difficult to be differentiated from dogs than trucks. To this end, we combine IDN with ASYM to generate more realistic label noise. Specifically, we choose similar images for each class and then we flip their labels to the next class.

3.A.2 Implementation Details

For backbone network, we use ResNet34 for MoCov2 and BYOL on CIFAR datasets. For projection MLP, we use 2-layer MLP with batch normalization [50] in the middle for MoCov2 and

Table 3.6: Test accuracy on CIFAR-10 datasets semantic label noise over different noise levels.

Dataset	TYPE-1					TYPE-2				
Noise Level	20%	40%	60%	80%	90%	10%	20%	30%	40%	45%
GCE	88.61±0.11	79.01±0.36	68.77±0.43	53.24±0.82	33.66±0.74	85.31±0.08	76.08±0.22	70.44±0.34	64.71±0.63	57.8±0.47
MoCo-GCE	**95.39**±0.04	**94.80**±0.10	**89.94**±0.16	**83.13**±0.95	**71.84**±0.59	**90.92**±0.05	**83.10**±0.12	**76.41**±0.17	**69.60**±0.29	**66.68**±0.55
BYOL-GCE	**94.86**±0.04	**93.69**±0.31	**89.19**±0.36	**83.53**±1.26	**75.77**±1.44	**89.64**±0.27	**83.34**±0.29	**75.65**±0.37	**69.32**±0.15	**66.86**±0.34
CT	91.06±0.33	72.78±0.35	44.30±0.47	25.37±0.18	17.30±0.23	85.52±0.12	76.11±0.12	65.36±0.20	55.54±0.81	48.64±0.38
MoCo-CT	**95.02**±0.43	**89.84**±0.67	**84.70**±0.96	**73.34**±1.84	**59.22**±3.11	**89.00**±0.45	**83.23**±0.77	**76.55**±0.92	**72.41**±1.53	**66.66**±1.98
BYOL-CT	**94.58**±0.07	**91.56**±0.42	**86.19**±0.70	**73.27**±0.61	**64.75**±1.95	**91.06**±0.47	**83.10**±0.20	**76.50**±0.18	**68.57**±1.23	**63.82**±1.58
ELR	94.19±0.05	91.75±0.51	82.72±0.43	70.86±0.46	39.05±0.72	86.69±0.02	79.06±0.08	71.02±0.11	62.09±0.27	58.02±0.26
MoCo-ELR	**95.76**±0.01	**94.96**±0.26	**84.30**±0.59	**79.99**±0.45	**63.63**±1.09	**88.27**±0.49	**84.53**±0.27	**78.45**±0.61	**77.57**±0.35	**69.27**±1.70
BYOL-ELR	**95.66**±0.02	**95.35**±0.07	**88.91**±0.31	**77.83**±0.42	**76.98**±0.60	**89.41**±0.87	**85.83**±0.70	**78.30**±0.55	**77.10**±0.62	**71.64**±1.50

BYOL. Since MoCov2 does not rely on the prediction MLP, we use 2-layer MLP with batch normalization in the middle for the prediction MLP of BYOL. The input and output dimensions for projection MLP and prediction MLP are set as 512. For ANIMAL-10N, we change the backbone network to ResNet50. In the meanwhile, the input and output dimensions for projection MLP and prediction MLP are set as 2048. The code for training MoCov2 and BYOL is adapted from `https://github.com/kibok90/imix`, where the data augmentations include both strong image augmentation from [13] and MixUp from [64]. MixUp [157] has a hyper-parameter λ controls the strength of interpolation between data points, where we set $\lambda = 1$ for CIFAR datasets and $\lambda = 2$ for ANIMAL-10. Once we trained the representation network, we train a linear classifier by different label noise methods on this frozen representation network.

The linear classifier is trained for 100 epochs using SGD, where the learning rate starts from {1, 5, 10, 20, 30} and it is reduced by a factor of 5 after 20, 30 and 40 epochs. For GCE method [161], its parameter q is selected from {0.2, 0.4, 0.6, 0.8, 0.9}; for Co-teaching method [38], the warmup parameters is selected from {5, 8, 10}; for ELR method [75], the parameter β is selected from {0.7, 0.9} and the parameter λ is selected from {3, 5, 7}.

3.A.3 Additional Results

Following [162, 12, 63], semantic label noise is a type of instance-dependent label noise which follows the intuition that hard instances are more likely to be mislabeled, where the hard instances are near the decision boundary of the model. To generate the semantic label noise, we train a VGG-13 [111] on training datasets for 30 epochs. Following [12], we select instances with highest mislabeling scores to corrupt. For the first case, we corrupt these instances with random labels. For the second case, we corrupt these instances with predictions of the model VGG-13. We term the former TYPE-1 label noise and the latter TYPE-2 label noise. The results for the two types of label noise are reported in Table 3.6. Therefore, extensive experiments have demonstrated the effectiveness of applying frozen SSL representations.

3.B Proofs for Theorem 3.3.1 and Proposition 3.4.1

3.B.1 Lemma 3.B.1

Lemma 3.B.1. *For any interval $I_\Delta \subset [-1.75, 2.25]$ with the length $\Delta = 4(1-e^{-d^{-5}})$, there exists at least one of those $\{\xi_i\}_{i=1}^n$ are within the interval I_Δ. When $n = \text{Poly}(d)$, the conclusion holds with probability at least $1 - e^{-d^5}$.*

Proof. Calculating event that there exist at least one of those $\{\xi_i\}_{i=1}^n$ are within the interval I_Δ is equivalent to calculate the event that all $\{\xi_i\}_{i=1}^n$ are not within the interval I_Δ. Specifically

$$\Pr[\text{At least one}] = 1 - \Pr[\text{None}].$$

Since all $\{\xi_i\}_{i=1}^n$ are sampled independently and we let $n = d^{10}$

$$\begin{aligned}\Pr[\text{None}] =& [p(\xi_i \notin I_\Delta)]^n \\ \leq& [\frac{4-\Delta}{4}]^n \\ =& [e^{-d^{-5}}]^n \\ =& e^{-d^5}.\end{aligned}$$

Then $\Pr[\text{At least one}] \geq 1 - e^{-d^5}$. □

3.B.2 Theorem 3.3.1

Proof. With the similar proof for $\{\zeta_i\}_{i=1}^n$, the conclusion of this Lemma also holds for $\{\zeta_i\}_{i=1}^n$.

As pointed out in [115], the gradient descent with logistic loss over a linearly separable data induces a maximum L2 margin solution. It indicates that the induced classifier separate the data with respect to noisy labels and also satisfies the maximum L2 margin. Specifically, $\tilde{\omega} = \frac{\omega^\star}{\|\omega^\star\|_2}$, where $\omega^\star$ is given by:

$$\omega^\star = \underset{\omega \in \mathbb{R}^d}{\text{argmin}} \|\omega\|^2 \text{ s.t. } \forall i : \omega^\top x_i \tilde{y}_i \geq 1. \tag{3.7}$$

The normalized optimal solution $\tilde{\omega} \in \{a_1 e_1 + a_2 e_2 : a_1 \in \mathbb{R}, a_2 \in \mathbb{R}\}$ since $a_3 e_3 + \cdots + a_d e_d$ is orthogonal to data point x_i for any $a_j \in \mathbb{R}$, $j \in \{3, 4, \cdots, d\}$ and any $i \in [n]$. Let $\tilde{\omega} = \tilde{a}_1 e_1 + \tilde{a}_2 e_2$.

By Lemma 3.B.1 with probability at least $1 - 2e^{-d^5}$, there exists a data point $(x_1 = e_1\zeta_1 + e_2\xi_2, \tilde{y}_1)$ where $|\zeta_1| \in [4-\Delta, 4], |\xi_1| \in [0, \Delta]$. We analyze the case where $\zeta_1 > 0, \xi_1 > 0$. The analysis for other cases $\zeta_1 < 0, \xi_1 > 0$, $\zeta_1 > 0, \xi_1 < 0$, and $\zeta_1 < 0, \xi_1 < 0$ are similar. If $\zeta_1 > 0, \xi_1 > 0$, then $\tilde{y}_1 = 1$.

Consider the worst data point with $\zeta_1 = 4 - \Delta$ and $\xi_1 = \Delta$ that decides the lowest prediction accuracy of the classifier. To classify this point into the cluster $\tilde{y} = 1$, we need at least $\tilde{a}_1 > \frac{-\Delta}{\sqrt{\Delta^2 + (\Delta-4)^2}}$ and $\tilde{a}_2 > \frac{4-\Delta}{\sqrt{\Delta^2+(\Delta-4)^2}}$. If there are other data points in this region, then the lower bound for $\tilde{a}_1$ is higher than $\frac{-\Delta}{\sqrt{\Delta^2+(\Delta-4)^2}}$ and the lower bound for $\tilde{a}_2$ is higher than $\frac{4-\Delta}{\sqrt{\Delta^2+(\Delta-4)^2}}$. Similarly, for other three cases, we have that $|\tilde{a}_1| < \frac{\Delta}{\sqrt{\Delta^2+(\Delta-4)^2}}$ and $\tilde{a}_2 > \frac{4-\Delta}{\sqrt{\Delta^2+(\Delta-4)^2}}$. Therefore,

area of region misclassified by the classifier $\tilde{\omega}$ is at most $14 + \frac{16\Delta}{4-\Delta}$ (the total area is 32). Since the joint distribution of $X \times Y$ is uniform over the support, then the probability that a data is misclassified (in the misclassified region) is at most $\frac{7}{16} + \frac{\Delta}{2(4-\Delta)}$, which is upper bounded by $\frac{7}{16} + \frac{\Delta}{6}$. Thus, the probability $\left|\Pr_{(x,y)}[\text{sign}(\tilde{\omega}^\top x) = y] - 0.5625\right| \leq \frac{2}{3}(1 - e^{-d^{-5}}) \leq \frac{2}{3d^5}$.

For self-supervised learning, as it is shown in [73], the minimal solution W_{SSL} to Eq. (3.1) is equivalent to the minimal solution to minimize $\left\|M - W^\top W\right\|_F^2$, where $M = \frac{1}{n}\sum_{i=1}^n x_i x_i^\top$. In the meanwhile, by Eckart-Young-Mirsky theorem, the span of W_{SSL} is the top 1 eigenvector of M. Let the top 1 eigenvector be $e = pe_1 + qe_1$ with $p^2 + q^2 = 1$. The corresponding eigenvalue is given by:

$$
\begin{aligned}
e^\top M e =& \frac{1}{n}\sum_{i=1}^n [(pe_1^\top + qe_2^\top)(\zeta_i e_1 + \xi_i e_2)(\zeta_i e_1^\top + \xi_i e_2^\top)(pe_1 + ne_2)] \\
=& \frac{1}{n}\sum_{i=1}^n [p^2\zeta_i^2 + q^2\xi_i^2 + 2pq\zeta_i\xi_i] \\
=& \frac{1}{n}\sum_{i=1}^n [(1-q^2)\zeta_i^2 + q^2\xi_i^2 + 2pq\zeta_i\xi_i] \qquad (p^2+q^2=1) \\
=& \frac{1}{n}\sum_{i=1}^n \zeta_i^2 - \frac{q^2}{n}\sum_{i=1}^n(\zeta_i^2 - \xi_i^2) + \frac{2pq}{n}\sum_{i=1}^n \zeta_i\xi_i \\
=& \frac{1}{n}\sum_{i=1}^n \zeta_i^2 - \frac{q^2}{n}\sum_{i=1}^n(\zeta_i^2 - \xi_i^2) + 2pq\mathbb{E}[\zeta\xi] + o(n^{-1/3}) \\
=& \frac{1}{n}\sum_{i=1}^n \zeta_i^2 - q^2 \underbrace{\frac{1}{n}\sum_{i=1}^n(\zeta_i^2 - \xi_i^2)}_{①} + o(n^{-1/3}).
\end{aligned}
$$

When n is large enough, ① converges to its expectation. Let $\bar{Z} = \frac{1}{n}\sum_{i=1}^n(\zeta_i^2 - \xi_i^2)$. By Hoeffding's inequality

$$\mathbb{P}\{|\bar{Z} - \mathbb{E}[\bar{Z}]| \geq 1\} \leq 2\exp\left(-\frac{2n}{16^2}\right),$$

where $\mathbb{E}[\bar{Z}] = \frac{63}{16} \approx 3.93$. With probability at least $1 - 2e^{-n/128}$, $\bar{Z} \in [\frac{47}{16}, \frac{79}{16}]$ and $\bar{Z} \gg o(n^{-1/3})$. Therefore, the eigenvalue corresponding to the eigenvector e is the largest when $q = 0$ and the top 1 eigenvector is $e = e_1$.

Without loss of generality, we let $W_{\text{SSL}} = e_1^\top$. After applying self-supervised transformation on inputs , the transformed data $\{(W_{\text{SSL}}x_i, \tilde{y}_i)\}_{i=1}^n = \{(y_i\zeta_i, \tilde{y}_i)\}_{i=1}^n$. We aim to learn any classifier where $\theta > 0$ to correctly predict all labels given inputs $\{y_i\zeta_i\}_{i=1}^n$ since $\zeta_i > 0, \forall i \in [n]$. The negative gradient of the logistic loss $\mathcal{L}(\theta)$ over $\{(y_i\zeta_i, \tilde{y}_i)\}_{i=1}^n$ is given by:

$$-\nabla_\theta\mathcal{L}(\theta) = \frac{1}{n}\sum_i^n \frac{\exp(-\tilde{y}_i y_i \zeta_i \theta)}{1 + \exp(-\tilde{y}_i y_i \zeta_i \theta)}\tilde{y}_i y_i \zeta_i.$$

Note that $\tilde{y}$ is independent with ζ. And $\tilde{y}_i y_i = 1$ with probability $9/16$ and $\tilde{y}_i y_i = -1$ with probability $7/16$. Then with high probability, $-\nabla_\theta\mathcal{L}(\theta) > 0$. Eventually, there is a unique

optimum $\theta = \theta_0 - \sum_i \alpha_i \nabla_\theta \mathcal{L}(\theta_i) > 0$ that minimizes the loss over $\{(y_i\zeta_i, \tilde{y}_i)\}_{i=1}^n$. When $\theta > 0$, the classifier gives the best decision boundary where $\text{sign}(\tilde{\theta} y_i \zeta_i) = y_i$. Hence, we have $\Pr_{(x,y)}[\text{sign}(\tilde{\theta}^\top W_{\text{SSL}} x) = y] \geq 1 - 2e^{-n/128}$. □

3.B.3 Proposition 3.4.1

Proof. The proof for Proposition 3.4.1 can be adapted from the proof for Theorem 3.3.1. The optimal W_{SSL} can be represented by the combination of e_1 and e_2. So we let $W_{\text{SSL}} = p_n e_1 + q_n e_2$, where p_n and q_n are optimal solutions depended on the sample size n. By the definition of convergence in probability, we need to show

$$\Pr[\|p_n e_1 + q_n e_2 - e_1\|_2 > \epsilon] \to 0 \text{ as } n \to \infty,$$

for every $\epsilon > 0$. When $p_n = 1$ as $n \to \infty$, the above condition holds since $p_n + q_n = 1$. From the proof of Theorem 3.3.1, we have

$$\Pr[p_n = 1] \geq 1 - 2e^{-n/128}.$$

Therefore

$$\begin{aligned}
\Pr[\|p_n e_1 + q_n e_2 - e_1\|_2 > \epsilon] =& \Pr[\|p_n e_1 + q_n e_2 - e_1\|_2 > \epsilon | p_n = 1] \Pr[p_n = 1] \\
&+ \Pr[\|p_n e_1 + q_n e_2 - e_1\|_2 > \epsilon | p_n \neq 1] \Pr[p_n \neq 1] \\
\leq& \Pr[\|p_n e_1 + q_n e_2 - e_1\|_2 > \epsilon | p_n = 1] + \Pr[p_n \neq 1] \\
\leq& 2e^{-n/128} \to 0 \text{ as } n \to \infty
\end{aligned}$$

□

3.C Proofs for Theorem 3.5.1

3.C.1 Gradient Descent Discussion

We discuss the behaviour of the linear classifier parameterized by ω_T given $-\widetilde{\nabla \mathcal{L}}(\omega_0)^\top \tilde{\mu} > \beta$, where T is the time to stop training, and $0 < \beta \leq 1$. We first introduce the following lemma, which is used to characterize the convexity of the logistic loss over the data.

Lemma 3.C.1. *Logistic loss $\mathcal{L}(\omega)$ is $\frac{\sigma_{\max}+1}{8}$-smooth when n is large enough.*

Proof. The derivative of $\mathcal{L}(\omega)$ is given by:

$$\nabla \mathcal{L}(\omega) = \frac{1}{n} \sum_i (1 - \frac{1}{1 + \exp(-\tilde{y}_i \omega^\top x_i)})(-\tilde{y}_i x_i).$$

Then, the hessian of $\mathcal{L}(\omega)$ is given by:

$$\nabla(\nabla \mathcal{L}(\omega)) = \frac{1}{n} \sum_{i=1}^n x_i \sigma(i)(1 - \sigma(i)) x_i^\top,$$

where $\sigma(i) = 1/(1+\exp(-\tilde{y}_i\omega^\top x_i))$, and $\sigma_{\max}$ is the largest eigenvalue of Σ. Since $x \sim \mathcal{N}(\mathbf{0}, \Sigma/2)$, and by the rates of convergence for law of large numbers, we have

$$\nabla^2\mathcal{L}(\omega) \geq \frac{1}{4}(\Sigma/2 + o(n^{-1/3})) \leq \frac{\sigma_{\max}+1}{8}.$$

□

Based on the properties of smoothness:

$$(\nabla\mathcal{L}(\omega_{t+1}) - \nabla\mathcal{L}(\omega_t))^\top(\omega_{t+1}-\omega_t) \leq L\|(\omega_{t+1}-\omega_t)\|_2^2,$$

where $L = \frac{\sigma_{\max}+1}{8}$. Given that the descent algorithm,

$$\omega_{t+1} = \omega_t - \alpha_t\nabla\mathcal{L}(\omega_t)$$

By choosing appropriate learning rate α_t (for example $\alpha_t = \frac{1}{2L}$), we then have

$$\nabla\mathcal{L}(\omega_{t+1})^\top\nabla\mathcal{L}(\omega_t) \geq \frac{1}{4L}\|\nabla\mathcal{L}(\omega_t))\|_2^2 > 0.$$

Or equivalently,

$$\widetilde{\nabla\mathcal{L}}(\omega_{t+1})^\top\widetilde{\nabla\mathcal{L}}(\omega_t) \geq \underbrace{\frac{\|\nabla\mathcal{L}(\omega_t))\|_2}{4L\|\nabla\mathcal{L}(\omega_{t+1}))\|_2}}_{\gamma_t} > 0.$$

For two unit vectors, $\widetilde{\nabla\mathcal{L}}(\omega_0)$ and $\tilde{\mu}$, the higher cosine similarity means the lower L2 distance:

$$\left\|-\widetilde{\nabla\mathcal{L}}(\omega_0) - \tilde{\mu}\right\|_2^2 = 2 + 2\widetilde{\nabla\mathcal{L}}(\omega_0)^\top\tilde{\mu} \leq 2(1-\beta).$$

Similarly,

$$\left\|\widetilde{\nabla\mathcal{L}}(\omega_{t+1}) - \widetilde{\nabla\mathcal{L}}(\omega_t)\right\|_2^2 = 2 - 2\widetilde{\nabla\mathcal{L}}(\omega_{t+1})^\top\widetilde{\nabla\mathcal{L}}(\omega_t) \leq 2(1-\gamma_t).$$

Therefore,

$$\begin{aligned}\left\|-\widetilde{\nabla\mathcal{L}}(\omega_1) - \tilde{\mu}\right\|_2^2 &= \left\|-\widetilde{\nabla\mathcal{L}}(\omega_1) + \widetilde{\nabla\mathcal{L}}(\omega_0) - \widetilde{\nabla\mathcal{L}}(\omega_0) - \tilde{\mu}\right\|_2^2 \\ &\leq 2\left\|\widetilde{\nabla\mathcal{L}}(\omega_1) - \widetilde{\nabla\mathcal{L}}(\omega_0)\right\|_2^2 + 2\left\|-\widetilde{\nabla\mathcal{L}}(\omega_0) - \tilde{\mu}\right\|_2^2 \\ &\leq 2(1-\gamma_0) + 2(1-\beta),\end{aligned}$$

Equivalently,

$$-\widetilde{\nabla\mathcal{L}}(\omega_1)^\top\tilde{\mu} \geq 2(\beta+\gamma_0-1).$$

This can be easily generalized the equation to time $t > 1$.

This can intuitively explain how larger $-\widetilde{\nabla\mathcal{L}}(\omega_0)^\top\tilde{\mu} > 0$ affects the performance of the classifier ω_T. The conclusion with more rigorous justification can be found in Theorem 1 in [75].

From another perspective, to intuitively understand why $-\widetilde{\nabla\mathcal{L}}(\omega_0)^\top\tilde{\mu} > \beta$ guarantees the behaviour of ω_T for $0 < \beta \leq 1$, we first decompose the gradient into two parts:

$$\begin{aligned}\nabla\mathcal{L}(\omega) =& \frac{1}{n}\sum_i (1 - \frac{1}{1+\exp(-\tilde{y}_i\omega^\top x_i)})(-\tilde{y}_i x_i)\\ =& \frac{1}{n}[\underbrace{\sum_{i\in I_c}(1 - \frac{1}{1+\exp(-\omega^\top x_i)})(-x_i)}_{\text{clean coefficients}} + \underbrace{\sum_{i\in I_n}(1 - \frac{1}{1+\exp(\omega^\top x_i)})(x_i)}_{\text{mislabeled coefficients}}],\end{aligned}$$

where the density of x is $\mathcal{N}(\mu, \Sigma)$, and I_c is the index set of clean examples and I_n is the index set of mislabeled examples. The first part is computed by weighted instances with clean labels, and we term the weights clean coefficients. Similarly, we term the weights for the second part mislabeled coefficients.

At the beginning ($t = 0$), clean samples dominate the gradient so we get the classifier closer to the optimal as $-\widetilde{\nabla\mathcal{L}}(\omega_0)^\top\tilde{\mu} > \beta$. Based on Proposition 5 in [75], the clean coefficients decrease and the mislabeled coefficients increase as the training progresses,. Eventually, they will achieve the balance, which leads to small $\nabla\mathcal{L}(\omega_t)$. Given that both the magnitude of $\nabla\mathcal{L}(\omega_t)$ and the learning rate α_t are small at time t, according to the gradient descent algorithm,

$$\omega_{t+1} = \omega_t - \alpha_t\nabla\mathcal{L}(\omega_t)$$

the learning will stop. Before that time, the learning is still dominated by clean examples and and the performance of the classifier improves until convergence.

3.C.2 Theorem 3.5.1

Proof. By the rates of convergence for law of large numbers

$$\begin{aligned}\nabla\mathcal{L}(\omega) =& \frac{1}{n}\sum_i (1 - \frac{1}{1+\exp(-\tilde{y}_i\omega^\top x_i)})(-\tilde{y}_i x_i)\\ =& \mathbb{E}[\nabla\mathcal{L}(\omega)] + o(n^{-1/3}).\end{aligned}$$

In this case of the symmetric label noise, the label noise function $\beta = -1$ with probability r and $\beta = 1$ with probability $1 - r$, where r controls the noise level.

We decompose the expected gradient into the following form

$$\begin{aligned}\mathbb{E}[\nabla\mathcal{L}(\omega_0)] =& \mathbb{E}[\mathbb{E}[\nabla\mathcal{L}(\omega_0)]|Y,\beta]\\ =& \frac{1-r}{2}\mathbb{E}[\nabla\mathcal{L}(\omega_0)|Y=1,\beta=1] + \frac{r}{2}\mathbb{E}[\nabla\mathcal{L}(\omega_0)|Y=1,\beta=-1]\\ &+ \frac{1-r}{2}\mathbb{E}[\nabla\mathcal{L}(\omega_0)|Y=-1,\beta=1] + \frac{r}{2}\mathbb{E}[\nabla\mathcal{L}(\omega_0)|Y=-1,\beta=-1],\end{aligned}$$

where the derivative of $\mathcal{L}(\omega)$ is given by:

$$\nabla\mathcal{L}(\omega) = \frac{1}{n}\sum_i (1 - \frac{1}{1+\exp(-\tilde{y}_i\omega^\top x_i)})(-\tilde{y}_i x_i). \tag{3.8}$$

To simplify the mathematical derivation, we assume that ω_0 is initialized at 0. Based on this, the expected gradient can be simplified:

$$\begin{aligned}\mathbb{E}[\nabla\mathcal{L}(\omega_0)] =& \frac{r}{2}\mathbb{E}[X|Y=1,\beta=-1] - \frac{1-r}{2}\mathbb{E}[X|Y=1,\beta=1] \\ =& (r-\frac{1}{2})\mathbb{E}[X].\end{aligned}$$

And,

$$\mathbb{E}[\nabla\mathcal{L}(\omega_0)^\top\mu] = (r-\frac{1}{2})\,\|\mu\|_2^2 \tag{3.9}$$

Then we compute $\|\nabla\mathcal{L}(\omega_0)\|_2$. By Jensen's inequality,

$$\begin{aligned}\|\nabla\mathcal{L}(\omega_0)\|_2 =& \frac{1}{2}\left\|\frac{1}{n}\sum_i \delta_i x_i\right\|_2 && \delta_i \text{ is either +1 or -1} \\ \leq& \frac{1}{2n}\sum_i \|x_i\|_2 \\ \leq& \frac{1}{2}\sqrt{\|\mu\|_2^2 + c\text{Trace}(\Sigma)},\end{aligned}$$

where the last inequality is by the concentration property of sub-gaussian random vector and $c > 0$ is a constant.

Since $-\nabla\mathcal{L}(\omega_0)^\top\mu > 0$ by Eq. (3.9) given a sufficient number of examples, the condition $-\frac{\nabla\mathcal{L}(\omega_0)^\top\mu}{\|\nabla\mathcal{L}(\omega_0)\|_2\|\mu\|_2}$ is then given by:

$$\begin{aligned}-\frac{\nabla\mathcal{L}(\omega_0)^\top\mu}{\|\nabla\mathcal{L}(\omega_0)\|_2\,\|\mu\|_2} =& \frac{(\frac{1}{2}-r)\,\|\mu\|_2^2 + o(n^{1/3})}{\|\nabla\mathcal{L}(\omega_0)\|_2\,\|\mu\|_2} \\ \geq& \frac{(1-2r)\,\|\mu\|_2}{\sqrt{\|\mu\|_2^2 + c\text{Trace}(\Sigma)}} + o(n^{1/3}).\end{aligned}$$

When the label noise is asymmetric, the results are the same though the mathematical expression is slightly different. For asymmetric label noise, we denote the label noise function $\beta(y) = -1$ with probability r when $y = -1$, $\beta(y) = 1$ with probability $1-r$ when $y = -1$, $\beta(y) = -1$ with probability $2r$ when $y = 1$, and $\beta(y) = 1$ with probability $1-2r$ when $y = 1$. Following similar derivations for the symmetric label noise,

$$\begin{aligned}\mathbb{E}[\nabla\mathcal{L}(\omega_0)] =& \mathbb{E}[\mathbb{E}[\nabla\mathcal{L}(\omega_0)]|Y,\beta] \\ =& \frac{1-r}{2}\mathbb{E}[\nabla\mathcal{L}(\omega_0)|Y=-1,\beta=1] + \frac{r}{2}\mathbb{E}[\nabla\mathcal{L}(\omega_0)|Y=-1,\beta=-1] \\ &+ \frac{1-2r}{2}\mathbb{E}[\nabla\mathcal{L}(\omega_0)|Y=1,\beta=1] + \frac{2r}{2}\mathbb{E}[\nabla\mathcal{L}(\omega_0)|Y=1,\beta=-1] \\ =& \frac{3r-1}{2}\mathbb{E}[X|Y=1].\end{aligned}$$

And,

$$\begin{aligned}\mathbb{E}[\nabla\mathcal{L}(\omega_0)^\top\mu] &= \frac{3r-1}{2}\Big[\int_{-\infty}^{+\infty}(\|\mu\|_2^2 + W)\,\mathrm{d}\mathbb{P}_W\Big] \\ &= \frac{3r-1}{2}\|\mu\|_2^2 .\end{aligned}$$

Therefore,

$$\begin{aligned}-\frac{\nabla\mathcal{L}(\omega_0)^\top\mu}{\|\nabla\mathcal{L}(\omega_0)\|_2\,\|\mu\|_2} &= \frac{(\frac{1}{2}-\frac{3r}{2})\|\mu\|_2^2 + o(n^{1/3})}{\|\mathbb{E}[\nabla\mathcal{L}(\omega_0)]\|_2\,\|\mu\|_2} \\ &\geq \frac{(1-3r)\|\mu\|_2}{\sqrt{\|\mu\|_2^2 + c\mathrm{Trace}(\Sigma)}} + o(n^{1/3}).\end{aligned}$$

□

3.D Proofs for Lemma 3.6.2, Lemma 3.6.3 and Proposition 3.6.4

3.D.1 Lemma 3.6.2

Proof. The function $-\log\mathbb{E}[e^t]$ is concave since for any t_1, t_2, and $\alpha \in [0,1]$

$$\begin{aligned}-\log\mathbb{E}[e^{\alpha t_1+(1-\alpha)t_2}] &= -\log\mathbb{E}[(e^{t_1})^\alpha(e^{t_2})^{1-\alpha}] \\ &= -\log\mathbb{E}[m_1^\alpha m_2^{1-\alpha}],\end{aligned}$$

where $m_1 = e^{t_1}, m_2 = e^{t_2}$. By Holder's inequality, the above equality is further given by

$$\begin{aligned}-\log\mathbb{E}[m_1^\alpha m_2^{1-\alpha}] &\geq -\log\left((\mathbb{E}[m_1])^\alpha(\mathbb{E}[m_2])^{1-\alpha}\right) \\ &= -\alpha\log\mathbb{E}[e^{t_1}] - (1-\alpha)\log\mathbb{E}[e^{t_2}].\end{aligned}$$

Therefore,

$$\begin{aligned}-\frac{1}{|J_i|}\sum_{k\in J_i}\log\mathbb{E}[e^{-\|f(u_k)-f(u_{g(k)})\|_2^2}] &\leq -\log\mathbb{E}[e^{-\frac{1}{|J_i|}\sum_{k\in J_i}\|f(u_k)-f(u_{g(k)})\|_2^2}] \\ &\leq -\log\mathbb{E}[e^{-\left\|\frac{1}{|J_i|}\sum_{k\in J_i}(f(u_k)-f(u_{g(k)}))\right\|_2^2}] \\ &\leq -\mathbb{E}[\log e^{-\left\|\frac{1}{|J_i|}\sum_{k\in J_i}(f(u_k)-f(u_{g(k)}))\right\|_2^2}] \\ &= \mathbb{E}\left[\left\|\frac{1}{|J_i|}\sum_{k\in J_i}(f(u_k)-f(u_{g(k)}))\right\|_2^2\right] \\ &= \mathbb{E}[\left\|\mu_i-\mu_j\right\|_2^2].\end{aligned}$$

Note that $-\|\cdot\|_2^2$ is concave and $-\log(\cdot)$ is convex. The proof is complete since

$$-\frac{1}{|J_i|}\sum_{k\in J_i}\log\mathbb{E}[e^{-\|f(u_k)-f(u_{g(k)})\|_2^2}] = -\frac{1}{|J_i|}\sum_{k\in J_i}\log\left(\mathcal{L}_{\mathrm{Uniform}}(k,g(k))\right).$$

□

3.D.2 Lemma 3.6.3

Proof. Let the region $R_{ij} = S_i \cap S_j$.

$$\begin{aligned}\mathbb{E}[\left\|f(u_i) - f(u_i^+)\right\|_2^2] =& \mathbb{E}[\left\|f(u_i) - f(u_i^+)\right\|_2^2 | u_i^+ \in R_{ij}] \Pr[u_i^+ \in R_{ij}] \\ &+ \mathbb{E}[\left\|f(u_i) - f(u_i^+)\right\|_2^2 | u_i^+ \notin R_{ij}] \Pr[u_i^+ \notin R_{ij}] \\ \geq& \delta \mathbb{E}[\left\|f(u_i) - f(u_i^+)\right\|_2^2 | u_i^+ \in R_{ij}] \end{aligned} \tag{3.10}$$

It shows that controlling the variance of a random variable controls the expected distance. For any different indices $i, j \in J_y$, with Eq.(3.10), we have for any $u_{ij} \in R_{ij}$

$$\begin{aligned}\mathbb{E}[\left\|f(u_i) - f(u_j)\right\|_2^2] \leq& 2\mathbb{E}[\left\|f(u_i) - f(u_{ij})\right\|_2^2] + 2\mathbb{E}[\left\|f(u_j) - f(u_{ij})\right\|_2^2] \\ \leq& \frac{2}{\delta}\mathbb{E}[\left\|f(u_i) - f(u_i^+)\right\|_2^2] + \frac{2}{\delta}\mathbb{E}[\left\|f(u_j) - f(u_j^+)\right\|_2^2], \end{aligned} \tag{3.11}$$

where we omit the subscriptions for expectation for simplicity when the context is clear.

The sample variance of the cluster y is given by

$$\widehat{\Sigma}_y = \frac{1}{|J_y|}\sum_{i \in J_y}(f(u_i) - \frac{\sum_{j \in J_y} f(u_j)}{|J_y|})(f(u_i) - \frac{\sum_{j \in J_y} f(u_j)}{|J_y|})^\top$$

By the property $\mathrm{Trace}(AB) = \mathrm{Trace}(BA)$,

$$\begin{aligned}\mathrm{Trace}(\mathbb{E}[\widehat{\Sigma}_y]) &= \frac{1}{|J_y|^3}\mathbb{E}\sum_{i \in J_y}\left\|\sum_{j \in J_y}(f(u_i) - f(u_j))\right\|_2^2 \\ &\leq \frac{1}{|J_y|^2}\mathbb{E}\sum_{i \in J_y}\sum_{j \in J_y}\left\|f(u_i) - f(u_j)\right\|_2^2 \\ &\leq \frac{2}{|J_y|\delta}\sum_{i \in J_y}\mathbb{E}[\left\|f(u_i) - f(u_i^+)\right\|_2^2] \qquad \text{by Eq. (3.11)} \\ &= \frac{2}{\delta|J_y|}\sum_{i \in J_y}\mathcal{L}_{\mathrm{Align}}(i)\end{aligned}$$

□

3.D.3 Proposition 3.6.4

Proof. We assume $|J_y|$ for all $y \in \mathcal{Y}$ are same as it is the most convenient and clear way to represent our results. We allow the different sizes for clusters and our results is unaffected. Note that $\log \mathbb{E}[e^t]$ is convex, following the spirits of proof from Lemma 3.6.2, we have

$$\log[\frac{1}{n(n-1)}\sum_{m \in \mathcal{Y}}\sum_{\substack{i,j \in J_m \\ i \neq j}}\mathcal{L}_{\mathrm{Uniform}}(i,j)] \geq -\frac{1}{n(n-1)}\sum_{m \in \mathcal{Y}}\sum_{\substack{i,j \in J_m \\ i \neq j}}\mathbb{E}_{\substack{u_i \sim P(u|x_i) \\ u_j \sim P(u|x_j)}}[\left\|f(u_i) - f(u_j)\right\|_2^2] \tag{3.12}$$

We replace terms in the right hand side of Eq.(3.12) associated with the the bound in Eq.(3.11), then we get

$$
\begin{aligned}
-\frac{1}{n(n-1)} \sum_{m\in\mathcal{Y}} \sum_{\substack{i,j\in J_m \\ i\neq j}} \mathbb{E}_{\substack{u_i\sim P(u|x_i) \\ u_j\sim P(u|x_j)}} [\left\|f(u_i)-f(u_j)\right\|_2^2] &\geq -\frac{2}{\delta n(n-1)} \sum_{m\in\mathcal{Y}} (|J_m-1|) \sum_{i\in J_m} \mathbb{E}[\left\|f(u_i)-f(u_i^+)\right\|_2^2] \\
&= -\frac{2(n/|\mathcal{Y}|-1)}{\delta n(n-1)} \sum_{m\in\mathcal{Y}} \sum_{i\in J_m} \mathbb{E}[\left\|f(u_i)-f(u_i^+)\right\|_2^2] \\
&= -\frac{2(n/|\mathcal{Y}|-1)}{\delta n(n-1)} \sum_{m\in\mathcal{Y}} \sum_{i\in J_m} \mathcal{L}_{\text{Align}}(i)
\end{aligned}
$$

□

Chapter 4

When Source-Free Domain Adaptation Meets Learning with Noisy Labels

4.1 Introduction

Deep learning demonstrates strong performance on various tasks across different fields. However, it is limited by the requirement of large-scale labeled and independent, and identically distributed (i.i.d) data. Unsupervised domain adaptation (UDA) is thus proposed to mitigate the distribution shift between the labeled source and unlabeled target domain. Due to the data privacy issue, accessing labeled source data is prohibitive. In view of the importance of data privacy, it is important to be able to adapt a source model to the unlabeled target domain without accessing the private source data.

The recent state-of-the-art SFDA methods [71, 147, 148] mainly focus on learning meaningful cluster structures in the feature space, and the quality of the learned cluster structures hinges on the reliability of pseudo labels generated by the source model. Among these methods, SHOT [71] purifies pseudo labels of target data based on nearest centroids, and then the purified pseudo labels are used to guide the self-training. G-SFDA [148] and NRC [147] encourage similar predictions to the data point and its neighbors. For a single target data point, when most of its neighbors are correctly predicted, these methods [71, 147, 148] can provide an accurate pseudo label to the data point. However, as we illustrate the problem in Figure 4.1(a-b), when the majority of its neighbors are incorrectly predicted to a category, it will be assigned with an incorrect pseudo label, misleading the learning of cluster structures. Since existing SFDA algorithms are not able to address this problem, the prediction error will accumulate as the training progresses.

In this chapter, we address the above problem by formulating the SFDA problem as *learning with label noise* (LLN). First, we show that there is a fundamental difference between the label noise in SFDA and conventional LLN scenarios. In conventional LLN scenarios studied in previous chapters, the label noise is generated by human annotators or image search engines [95, 141, 138], where the underlying distribution assumption is that the mislabeling rate for a sample is bounded. However, in the SFDA scenarios, the label noise is generated by the source model, where we prove that the mislabeling rate for a sample can approach 1. We term the former label noise in LLN as *bounded label noise* and the latter label noise in SFDA as *un-*

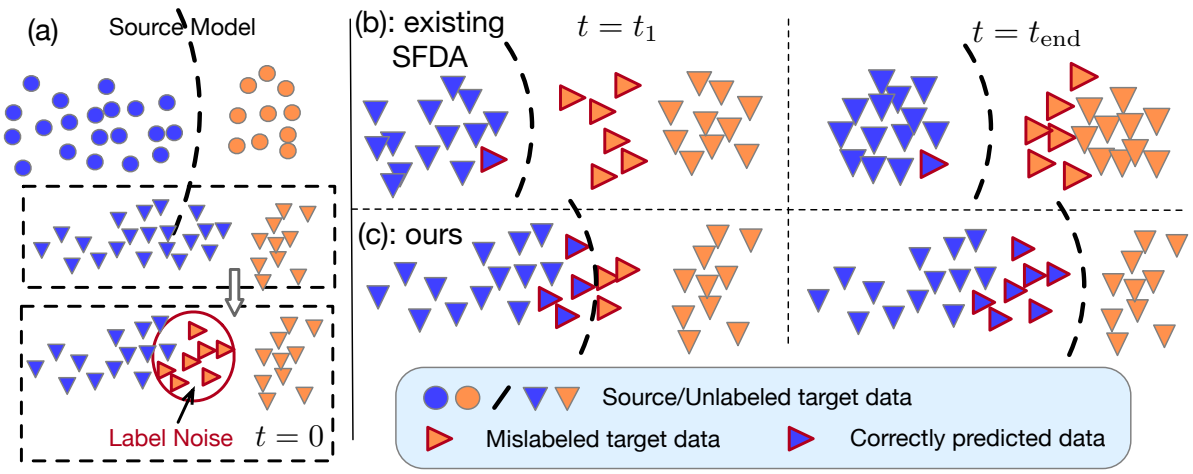


Figure 4.1: Overview of the SFDA problem and our method. (a) The SFDA problem can be formulated as an LLN problem. (b) The existing SFDA algorithms [71, 147, 148] leveraging the local cluster information cannot address label noise due to the unbounded label noise (see Section 4.3 for details). (c) We prove that ETP exists in SFDA, which can be leveraged to address the unbounded label noise (see Section 4.4 for details).

bounded label noise. Moreover, we theoretically show that most existing LLN methods, which rely on bounded label noise assumption, are unable to address the label noise in SFDA due to the fundamental difference.

To this end, we propose to leverage an observation in SFDA that classifiers can successfully predict mislabeled samples with high accuracy during the early-time training stage, which is termed as *early-time training phenomenon* (ETP) [75], for addressing the unbounded label noise to improve the efficiency of existing SFDA algorithms. Although ETP has been previously observed in [75], it has only been studied in the bounded random label noise in the conventional LLN scenarios. The bounded random label noise is a common assumption in the conventional LLN scenario [75, 161, 134, 95], but it is not realistic in the SFDA scenario. In SFDA, we rigorously establish using a high-dimensional Gaussian model that ETP also exists in the unbounded label noise scenario. Hence we can empirically justify that by leveraging ETP, existing SFDA algorithms can be substantially improved by embedding a simple regularization term into the SFDA objective functions. For this purpose, we choose ELR [75] from the conventional LLN domain as the regularization term to improve SFDA algorithms. As a comparison, we also apply other existing LLN methods, including GCE [161], SL [134], GJS [24] and PLC [160], to SFDA. Our empirical evidence shows that they are inappropriate for addressing the label noise in SFDA. This is also consistent with our theoretical results.

The main contributions of this chapter are summarized as follows.

- We establish the connection between the SFDA domain and the LLN domain. Compared with the conventional LLN problem that assumes bounded label noise, the problem in SFDA can be viewed as the problem of LLN with the unbounded label noise.
- We theoretically and empirically justify that ETP exists in the unbounded label noise scenario.

- We conduct extensive experiments to show that ETP can be utilized to improve many existing SFDA algorithms by a large margin across many SFDA benchmark datasets.

4.2 Related Work And Problem Setting

In this section, we first briefly discuss some work related to domain adaptation. Then we introduce some notations and the SFDA problem setting. The literature review of learning with label noise can be found in Chapter 1.

4.2.1 Related Work

Unsupervised domain adaptation. UDA has been extensively studied in the past, and the main idea is to mitigate the distribution discrepancy of different domains. Concretely, existing methods follow two schemes: explicit distribution distance alignment and adversarial training. Explicit distribution distance alignment calculates an explicit statistical distance between two domains such as the maximum mean discrepancy (MMD) [80, 127], the Wasserstein distance [110, 62], and the contrastive similarities [54]. Adversarial training implicitly aligns distributions of different domains by playing a min-max game through a domain discriminator as done in DANN [28]. Various designs have been proposed to estimate the domain discrepancy by domain discriminator or task classifiers [108, 79, 158, 45, 131]. However, the above approaches generally require access to source data which is not applicable in source-free domain adaptation.

Source-free domain adaptation. Recently, SFDA are studied for data privacy. The first branch of research is to leverage the target pseudo labels to conduct self-training to implicitly achieve adaptation [72, 123, 1, 148]. SHOT [71] introduces k-means clustering and mutual information maximization strategy for self-training. NRC [147] further investigates the neighbors of target clusters to improve the accuracy of pseudo labels. The other branch is to utilize the generative model to synthesize target-style training data [100, 77]. Some methods also explore the SFDA algorithms in various settings. USFDA [59] and FS [60] design methods for universal and open-set UDA. In this chapter, we regard SFDA as the LLN problem, and we aim to ameliorate the label noise to improve SFDA algorithms.

4.2.2 Problem Setting

Given n source data points $\{(x_i^s, y_i^s)\}_{i=1}^n$ sampled from the source distribution $\mathcal{D}_S$ and m unlabeled target data points $\{x_j^t\}_{i=1}^m$ sampled from the target distribution $\mathcal{D}_T$. For the source-free domain adaptation problem, the goal is to train a target classifier f_T to predict accurate labels for the target domain. Due to the data privacy issue, we cannot access to the source data. Instead, we have the source classifier f_S trained on the source data to minimizing the cross-entropy loss. Thus, the target classifier f_T can only be learned by adapting from the source classifier f_S with the help of some unlabeled data $\{x_j^t\}_{i=1}^m$. In this chapter, we relate this classic SFDA problem to the LLN problem, and we propose to solve it by LLN methods.

4.3 Label Noise In SFDA

The presence of label noise on training datasets has been shown to degrade the model performance [88, 38]. In SFDA, existing algorithms rely on pseudo-labels produced by the source model, which are inevitably noisy due to the domain shift. The SFDA methods such as [71, 147, 148] cannot tackle the situation when some target samples and their neighbors are all incorrectly predicted by the source model. We formulate the SFDA as the problem of LLN to address this issue. We show that the label noise in SFDA is unbounded. In contrast, the label noise in conventional LLN domain is bounded.

We first analyze the fundamental difference between the label noise in the SFDA and that in conventional LLN scenarios. Then, we prove that most existing LLN methods that rely on the bounded assumption cannot address the label noise in SFDA due to the difference.

Label noise in conventional LLN settings: In conventional label noise settings, the injected noisy labels are collected by either human annotators or image search engines [65, 70, 141]. The label noise is usually assumed to be either independent of instances (i.e. symmetric label noise or asymmetric label noise) [95, 76, 145] or dependent of instances (i.e. instance-dependent label noise) [8, 139]. The underling assumption for them is that a sample $\mathbf{x}$ has the highest probability of being in the correct class y, i.e. $\Pr[\tilde{Y} = i|Y = i, X = x] > \Pr[\tilde{Y} = j|Y = i, X = x],\ \forall x \in \mathcal{X}, i \neq j$, where $\tilde{Y}$ is the noisy label and Y is the ground-truth label for input X. Equivalently, it assumes a bounded noise rate. For example, given an image to annotate, the mislabeling rate for the image is bounded by a small number, which is realistic in conventional LLN settings [139, 18]. When the label noise is generated by the source model the underlying assumption of these types of label noise do not hold.

Label noise in SFDA settings: As for the label noise generated by the source model, the quality of generated labels is inherently controlled by the magnitude of the distribution shift between two domains. When the distribution shift is larger, the mislabeling error is larger. More importantly, the mislabeling rate for an image can approach 1, that is, $\Pr[\tilde{Y} = j|Y = i, X = x] \rightarrow 1,\ \exists \mathcal{S} \subset \mathcal{X},\ \forall x \in \mathcal{S}, i \neq j$.

First, we we focus on explaining the relationship between the label noise and the distribution shift. We consider a two-component Gaussian mixture distribution with equal priors for both domains. Let the first component ($y = 1$) of the source domain distribution $\mathcal{D}_S$ be $\mathcal{N}(\boldsymbol{\mu}_1, \sigma^2\mathbf{I}_d)$, and the second component ($y = -1$) of $\mathcal{D}_S$ be $\mathcal{N}(\boldsymbol{\mu}_2, \sigma^2\mathbf{I}_d)$. For the target domain distribution $\mathcal{D}_T$, let the first component ($y = 1$) of $\mathcal{D}_T$ be $\mathcal{N}(\boldsymbol{\mu}_1 + \boldsymbol{\Delta}, \sigma^2\mathbf{I}_d)$, and the second component ($y = -1$) of $\mathcal{D}_S$ be $\mathcal{N}(\boldsymbol{\mu}_2 + \boldsymbol{\Delta}, \sigma^2\mathbf{I}_d)$, where $\boldsymbol{\Delta} \in \mathbb{R}^d$ is the shift of the two domains. The underlying assumption for the two domains is the covariate shift assumption, which is the most commonly used one that assumes the conditional probability of labels given instances for two domains is unchanged during the domain shift [35, 119]. The following theorem characterizes the relationship between the labeling error and the distribution shift.

Remark: Here we choose two Gaussian distributions with equivalent variance to deliver our theoretical results. However, the results are not restricted to these limitations. We can have have Gaussian distributions with different variance or even t-distribution. The reason for choosing Gaussian is that it is simple and straightforward to deliver our results.

Theorem 4.3.1. *Without loss of generality, we assume that the $\boldsymbol{\Delta}$ is positively correlated with*

the vector $\boldsymbol{\mu_2} - \boldsymbol{\mu_1}$. *Let* f_S *be the Bayes optimal classifier for the source domain* S. *Then*

$$\Pr_{(\mathbf{x},y)\sim\mathcal{D}_T}[f_S(\mathbf{x}) \neq y] = \frac{1}{2}\Phi(-\frac{d_1}{\sigma}) + \frac{1}{2}\Phi(-\frac{d_2}{\sigma}), \tag{4.1}$$

where $d_1 = \left\|\frac{\boldsymbol{\mu_2}-\boldsymbol{\mu_1}}{2} - \mathbf{c}\right\| \text{sign}(\left\|\frac{\boldsymbol{\mu_2}-\boldsymbol{\mu_1}}{2}\right\| - \|\mathbf{c}\|)$, $d_2 = \left\|\frac{\boldsymbol{\mu_2}-\boldsymbol{\mu_1}}{2} + \mathbf{c}\right\|$, $\mathbf{c} = (\boldsymbol{\mu_2} - \boldsymbol{\mu_1})\frac{\boldsymbol{\Delta}^\top(\boldsymbol{\mu_2}-\boldsymbol{\mu_1})}{\|\boldsymbol{\mu_2}-\boldsymbol{\mu_1}\|^2}$, *and* Φ *is the standard normal cumulative distribution function.*

The proof is provided in 4.A. Theorem 4.3.1 indicates that the labeling error for the target domain can be represented by a function of the domain shift $\boldsymbol{\Delta}$. The projection of the domain shift $\boldsymbol{\Delta}$ on the vector $\boldsymbol{\mu_2} - \boldsymbol{\mu_1}$ is given by $\mathbf{c}$. We note that the mislabeling error is degraded to the Bayes error when the source and target domains are the same. Specifically, the labeling error in Eq. (4.1) is minimized when $\mathbf{c} = \mathbf{0}$, which is the Bayes error and the error cannot be reduced [27]. Since $\mathbf{c}$ is on the direction of $\boldsymbol{\mu_2} - \boldsymbol{\mu_1}$, $\mathbf{c}$ can also be represented by $a(\boldsymbol{\mu_2} - \boldsymbol{\mu_1})$, where $a \in \mathbb{R}$ characterizes the magnitude of the domain shift. By taking the gradient on the Eq. (4.1) with respect to α, we can find that the labeling error increases when α increases. It implies the larger the domain shift, the greater the mislabeling error. We defer the proof and details to Appendix 4.A.

While Theorem 4.3.1 shows that the label noise is inherently controlled by $\boldsymbol{\Delta}$, the following theorem characterizes that the label noise is unbounded.

Theorem 4.3.2. *Without loss of generality, we assume that the* $\boldsymbol{\Delta}$ *is positively correlated with the vector* $\boldsymbol{\mu_2} - \boldsymbol{\mu_1}$. *For* $(\mathbf{x}, y) \sim \mathcal{D}_T$, *if* $\mathbf{x} \in \mathbf{R}$, *then*

$$\Pr[f_S(\mathbf{x}) \neq y] \geq 0.99, \tag{4.2}$$

where $\mathbf{R} = \mathbf{R}_1 \bigcap \mathbf{R}_2$, $\mathbf{R}_1 = \{\mathbf{x} : \left\|\mathbf{x} - \boldsymbol{\mu}_1 - \boldsymbol{\Delta}\right\| \leq \sigma(\frac{\sqrt{d}}{2} - \frac{\log 99}{\sqrt{d}})\}$, *and* $\mathbf{R}_2 = \{\mathbf{x} : \mathbf{x}^\top \mathbf{1}_d > (\sigma d + 2\boldsymbol{\mu}_1^\top \mathbf{1}_d)/2\}$. *Meanwhile,* $\mathbf{R}$ *is non-empty when* $\alpha > (\log 99)/d$, *where* $\alpha = \frac{\boldsymbol{\Delta}^\top(\boldsymbol{\mu_2}-\boldsymbol{\mu_1})}{\|\boldsymbol{\mu_2}-\boldsymbol{\mu_1}\|^2} > 0$ *is the magnitude of the domain shift along the direction* $\boldsymbol{\mu_2} - \boldsymbol{\mu_1}$.

The proof is provided in 4.B. Conventional LLN methods assume that the label noise is bounded $\Pr[f_H(\mathbf{x}) \neq y] < c_0$, $\forall(\mathbf{x}, y) \sim \mathcal{D}_T$, where f_H is the labeling function, and $c_0 = 0.5$ if the number of clean samples of each component are the same [18]. However, Theorem 4.3.2 indicates that the label noise generated by the source model is unbounded for any $\mathbf{x} \in \mathbf{R}$. In practice, region $\mathbf{R}$ is non-empty as neural networks are usually trained on high dimensional data such that $d \gg 1$, so $\alpha > (\log 99)/d \to 0$ is easy to satisfy. The probability measure on $\mathbf{R} = \mathbf{R}_1 \bigcap \mathbf{R}_2$ (i.e. $\Pr_{(\mathbf{x},y)\sim\mathcal{D}_T}[\mathbf{x} \in \mathbf{R}]$) increases as the magnitude of the domain shift α increases, meaning more data points contradict the conventional LLN assumption. More details about this can be found in Appendix 4.B.

Given that the unbounded label noise exists in SFDA, the following Lemma establishes that many existing LLN methods [134, 89, 24, 85], which rely on the bounded assumption, are *not* noise tolerant in SFDA.

Lemma 4.3.3. *Let the risk of the function* $h : \mathcal{X} \to \mathcal{Y}$ *under the clean data be* $R(h) = \mathbb{E}_{\mathbf{x},y}[\ell_{LLN}(h(\mathbf{x}), y)]$, *and the risk of* h *under the noisy data be* $\widetilde{R}(h) = \mathbb{E}_{\mathbf{x},\tilde{y}}[\ell_{LLN}(h(\mathbf{x}), \tilde{y})]$, *where the noisy data follows the unbounded assumption, i.e.* $\Pr[\tilde{y} \neq y|\mathbf{x} \in \mathbf{R}] = 1$ *for a subset* $\mathbf{R} \subset \mathcal{X}$. *Then the global minimizer* $\tilde{h}^\star$ *of* $\widetilde{R}(h)$ *disagrees with the global minimizer* $h^\star$ *of* $R(h)$ *on data points* $\mathbf{x} \in \mathbf{R}$.

The proof is provided in 4.C. We denote ℓ_{LLN} by the existing LLN methods in [134, 89, 24, 85]. When the noisy data follows the bounded assumption, these methods are noise tolerant as the minimizer $\tilde{h}^\star$ converges to the minimizer $h^\star$. We defer the proof and details to Appendix 4.C.

4.4 Learning With Label Noise in SFDA

Given the fundamental difference between the label noise in SFDA and the label noise in conventional LLN scenarios, existing LLN methods, whose underlying assumption is bounded label noise, cannot be applied to solve the label noise in SFDA. This section focuses on investigating how to address the unbounded label noise in SFDA.

Motivated by the recent studies [75, 4], which observed an early-time training phenomenon (ETP) on noisy datasets with bounded random label noise. We find that ETP does not rely on the bounded random label noise assumption, and it can be generalized to the unbounded label noise in SFDA. ETP describes the training dynamics of the classifier that has higher prediction accuracy for mislabeled samples during the early-training stage. To theoretically prove ETP in the presence of unbounded label noise, we first describe the problem setup.

We still consider a two-component Gaussian mixture distribution with equal priors. We denote y by the true label for $\mathbf{x}$, and assume it is a balanced sample from $\{-1, +1\}$. The instance $\mathbf{x}$ is sampled from the distribution $\mathcal{N}(y\boldsymbol{\mu},\ \sigma\mathbf{1}_d)$, where $\|\boldsymbol{\mu}\| = 1$. We denote $\tilde{y}$ by the noisy label for $\mathbf{x}$. We observe that the label noise generated by the source model is close to the decision boundary revealed in Theorem 4.3.2. So, we let $\tilde{y} = y\beta(\mathbf{x}, y)$, where $\beta(\mathbf{x}, y) = \text{sign}(\mathbb{1}\{y\mathbf{x}^\top\boldsymbol{\mu} > r\} - 0.5)$ is the label flipping function, and r controls the mislabeling rate. If $\beta(\mathbf{x}, y) < 1$, then the data point $\mathbf{x}$ is mislabeled. Meanwhile, the label noise is unbounded by adopting the label flipping function $\beta(\mathbf{x}, y)$: $\Pr[\tilde{y} \neq y | y\mathbf{x}^\top\boldsymbol{\mu} \leq r] = 1$, where $\mathbf{R} = \{\mathbf{x} : y\mathbf{x}^\top\boldsymbol{\mu} \leq r\}$.

We study the early-time training dynamics of gradient descent on the linear classifier. The parameter θ is learned over the unbounded label noise data $\{x_i, \tilde{y}_i\}_{i=1}^n$ with the following logistic loss function:

$$\mathcal{L}(\theta_{t+1}) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + \exp\left(-\tilde{y}_i\theta_{t+1}^\top \mathbf{x}_i\right)\right),$$

where $\theta_{t+1} = \theta_t - \eta\nabla_\theta\mathcal{L}(\theta_t)$, and η is the learning rate. Then the following theorem builds the connection between the prediction accuracy for mislabeled samples at an early-training time T.

Theorem 4.4.1. *Let $B = \{\mathbf{x} : \tilde{y} \neq y\}$ be a set of mislabeled samples. Let $\kappa(B; \theta)$ be the prediction accuracy calculated by the ground-truth labels and the predicted labels by the classifier with parameter θ for mislabeled samples. If at most half of the samples are mislabeled ($r < 1$), then there exists a proper time T and a constant $c_0 > 0$ such that for any $0 < \sigma < c_0$ and $n \to \infty$, with probability $1 - o_p(1)$:*

$$\kappa(B; \theta_T) \geq 1 - \exp\{-\frac{1}{200}g(\sigma)^2\}, \tag{4.3}$$

where $g(\sigma) = \frac{\text{Erf}[\frac{1-r}{\sqrt{2}\sigma}]}{2(1+2\sigma)\sigma} + \frac{\exp(-\frac{(r-1)^2}{2\sigma^2})}{\sqrt{2\pi}(1+2\sigma)} > 0$ is a monotone decreasing function that $g(\sigma) \to \infty$ as $\sigma \to 0$, and $\text{Erf}[x] = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}\,\mathrm{d}t$.

The proof is provided in Appendix 4.D. Compared to ETP found in [75], where the label noise is assumed to be bounded, Theorem 4.4.1 presents that ETP also exists even though the label noise is unbounded. At a proper time T, the classifier trained by the gradient descent algorithm can provide accurate predictions for mislabeled samples, where its accuracy is lower bounded by a function of the variance of clusters σ. When $\sigma \to 0$, the predictions of all mislabeled samples equal to their ground-truth labels (i.e. $\kappa(B;\theta_T) \to 1$). When the classifier is trained for a sufficiently long time, it will gradually memorize mislabeled data. The predictions of mislabeled samples are equivalent to their incorrect labels instead of their ground-truth labels [75, 87]. Based on these insights, the memorization of mislabeled data can be alleviated by leveraging their predicted labels during the early-training time.

Theorem 4.4.1 rigorously justifies that the classifier can also have higher prediction accuracy for mislabeled samples with the unbounded label noise at proper epoch number T. To leverage the predictions during the early-training time, we adopt a recently established method, early learning regularization (ELR) [75], which encourages model predictions to stick to the early-time predictions for $\mathbf{x}$. Since ETP exists in the scenarios of the unbounded label noise, ELR can be applied to solve the label noise in SFDA. The regularization is given by:

$$\mathcal{L}_{\text{ELR}}(\theta_t) = \log(1 - \bar{y}_t^\top f(\mathbf{x};\theta_t)), \tag{4.4}$$

where we overload $f(\mathbf{x};\theta_t)$ to be the probabilistic output for the sample $\mathbf{x}$, and $\bar{y}_t = \beta\bar{y}_{t-1} + (1-\beta)f(\mathbf{x};\theta_t)$ is the moving average prediction for $\mathbf{x}$, where β is a hyperparameter. To see how ELR prevents the model from memorizing the label noise, we calculate the gradient of Eq. (4.4) with respect to $f(\mathbf{x};\theta_t)$, which is given by:

$$\frac{\mathrm{d}\mathcal{L}_{\text{ELR}}(\theta_t)}{\mathrm{d}f(\mathbf{x};\theta_t)} = -\frac{\bar{y}_t}{1 - \bar{y}_t^\top f(\mathbf{x};\theta_t)}.$$

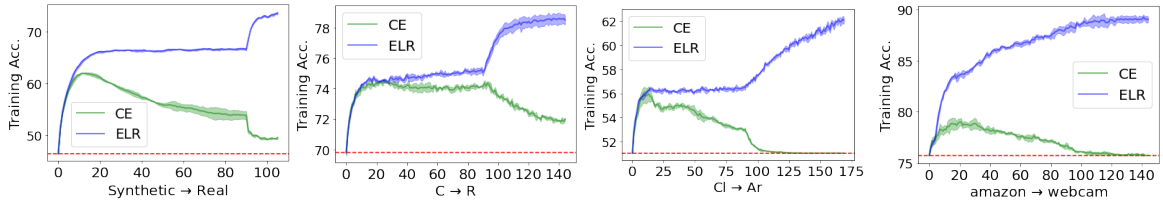
Note that minimizing Eq. (4.4) forces $f(\mathbf{x};\theta_t)$ to close to $\bar{y}_t$. When $\bar{y}_t$ is aligned better with $f(\mathbf{x};\theta_t)$, the magnitude of the gradient becomes larger. It makes the gradient of aligning $f(\mathbf{x};\theta_t)$ with $\bar{y}_t$ overwhelm the gradient of other loss terms that align $f(\mathbf{x};\theta_t)$ with noisy labels. As the training progresses, the moving averaged predictions $\bar{y}_t$ for target samples gradually approach their ground-truth labels till the time T. Therefore, Eq. (4.4) prevents the model from memorizing the label noise by forcing the model predictions to stay close to these moving averaged predictions $\bar{y}_t$, which are very likely to be ground-truth labels.

Some existing LLN methods propose to assign pseudo labels to data or require two-stage training for label noise [18, 164, 160]. Unlike these LLN methods, Eq. (4.4) can be easily embedded into any existing SFDA algorithms without conflict. The overall objective function is given by:

$$\mathcal{L} = \mathcal{L}_{\text{SFDA}} + \lambda\mathcal{L}_{\text{ELR}}, \tag{4.5}$$

where $\mathcal{L}_{\text{SFDA}}$ is *any* SFDA objective function, and λ is a hyperparameter.

Empirical Observations on Real-World Datasets. We empirically verify that target classifiers have higher prediction accuracy for target data during the early training stage. We propose leveraging this benefit to prevent the classifier from memorizing the noisy labels. The observations are shown in Figure 4.2. The parameters of classifiers are initialized by source models.



(a) VisDA-C (b) DomainNet (c) Office-Home (d) Office-31

Figure 4.2: Training accuracy on various target domains. The source models initialize the classifiers and annotate unlabeled target data. As the classifiers memorize the unbounded label noise very fast, we evaluate the prediction accuracy on target data every batch for the first 90 steps. After the 90 steps, we evaluate the prediction accuracy for every 0.3 epoch. We use the CE and ELR to train the classifiers on the labeled target data, shown in solid green lines and solid blue lines, respectively. The dotted red line represents the accuracy of labeling target data. Eventually, the classifiers memorize the label noise, and the prediction accuracy equals the labeling accuracy (shown in (c-d)). Additional results on transfer pairs can be found in Appendix 4.E.

Labels of target data are annotated by the initialized classifiers. We train the target classifiers on target data with the standard cross-entropy (CE) loss. The solid green lines represent the training accuracy of optimizing the classifiers with CE loss, while the solid blue lines represent that with ELR loss. The dotted red lines represent the labeling accuracy of the initialized classifiers. Considering that the classifiers memorize the unbounded label noise very fast, we evaluate the prediction accuracy on target data every batch for the first 90 steps. After 90 steps, we evaluate the prediction accuracy for every 0.33 epoch. The green lines show that ETP exists in SFDA, which is consistent with our theoretical result. Meanwhile, in most scenarios (3 out of 4 datasets), green lines also show that classifiers provide higher prediction accuracy during the first a few iterations. After a few iterations, they start to memorize the label noise. Eventually, the classifiers are expected to memorize the whole datasets. For conventional LLN settings, it has been empirically verified that it takes a *much longer* time before classifiers start memorizing the label noise [75, 138]. We provide further analysis in Appendix 4.G. We highlight that PCL [160] leverages ETP at every epoch, so it cannot capture the benefits of ETP and is inappropriate for unbounded label noise due to the fast memorization speed in SFDA. As a comparison, we choose ELR since it leverages ETP at every batch. The blue lines show that leveraging ETP via ELR can address the memorization of noisy labels in SFDA.

4.5 Experiments

We aim to improve the efficiency of existing SFDA algorithms by using ELR to leverage ETP. We evaluate the performance on four different SFDA benchmark datasets: Office-31 [106], Office-Home [128], VisDA [97] and DomainNet [96]. Due to the limited space, the results on the dataset Office-31 and additional experimental details are provided in Appendix 4.F.

Evaluation. We incorporate ELR into three existing baseline methods: SHOT [71], G-SFDA [161], and NRC [147]. SHOT uses k-means clustering and mutual information maximization strategy to train the representation network while freezing the final linear layer. G-

Table 4.1: Accuracies (%) on Office-31 for ResNet50-based methods.

Method	SF	A→D	A→W	D→W	W→D	D→A	W→A	Avg
MCD [108]	✗	92.2	88.6	98.5	**100.0**	69.5	69.7	86.5
CDAN [79]	✗	92.9	94.1	98.6	**100.0**	71.0	69.3	87.7
MDD [158]	✗	90.4	90.4	98.7	99.9	75.0	73.7	88.0
BNM [20]	✗	90.3	91.5	98.5	**100.0**	70.9	71.6	87.1
DMRL [137]	✗	93.4	90.8	99.0	**100.0**	73.0	71.2	87.9
BDG [146]	✗	93.6	93.6	99.0	**100.0**	73.2	72.0	88.5
MCC [52]	✗	95.6	95.4	98.6	**100.0**	72.6	73.9	89.4
SRDC [122]	✗	95.8	95.7	99.2	**100.0**	76.7	77.1	90.8
RWOT [144]	✗	94.5	95.1	**99.5**	**100.0**	**77.5**	77.9	90.8
RSDA-MSTN [37]	✗	**95.8**	**96.1**	99.3	**100.0**	77.4	**78.9**	**91.1**
Source Only	✓	80.8	76.9	95.3	98.7	60.3	63.6	79.3
+ELR	✓	90.9	89.0	98.2	**100.0**	67.1	64.1	84.9
SHOT [71]	✓	94.0	90.1	98.4	99.9	74.7	74.3	88.6
+ELR	✓	94.9	91.6	98.7	**100.0**	75.2	74.5	89.3
G-SFDA [148]	✓	85.9	87.3	98.6	99.8	71.4	72.1	85.8
+ELR	✓	86.9	87.8	98.7	99.8	71.4	72.9	86.2
NRC [147]	✓	93.7	93.8	97.8	**100.0**	75.5	75.6	89.4
+ELR	✓	93.8	93.3	98.0	**100.0**	76.2	76.9	89.6

SFDA aims to cluster target data with similar neighbors and attempts to maintain the source domain performance. NRC also explores the neighbors of target data by graph-based methods. ELR can be easily embedded into these methods by simply adding the regularization term into the loss function to optimize without affecting existing SFDA frameworks. We average the results based on three random runs.

Results. Tables 4.1-4.4 show the results before/after leveraging the early-time training phenomenon, where Table 4.1 is shown in Appendix 4.F. Among these tables, the top part shows the results of conventional UDA methods, and the bottom part shows the results of SFDA methods. In the tables, we use SF to indicate whether the method is source free or not. We use Source Only + ELR to indicate ELR with self-training. The results show that ELR itself can boost the performances. As existing SFDA methods are not able to address unbounded label noise, incorporating ELR into these SFDA methods can further boost the performance. The four datasets, including all 31 pairs (e.g., $A \rightarrow D$) of tasks, show better performance after solving the unbounded label noise problem using the early-time training phenomenon. Meanwhile, solving the unbounded label noise on existing SFDA methods achieves state-of-the-art on all benchmark datasets. These SFDA methods also outperform most methods that need to access source data.

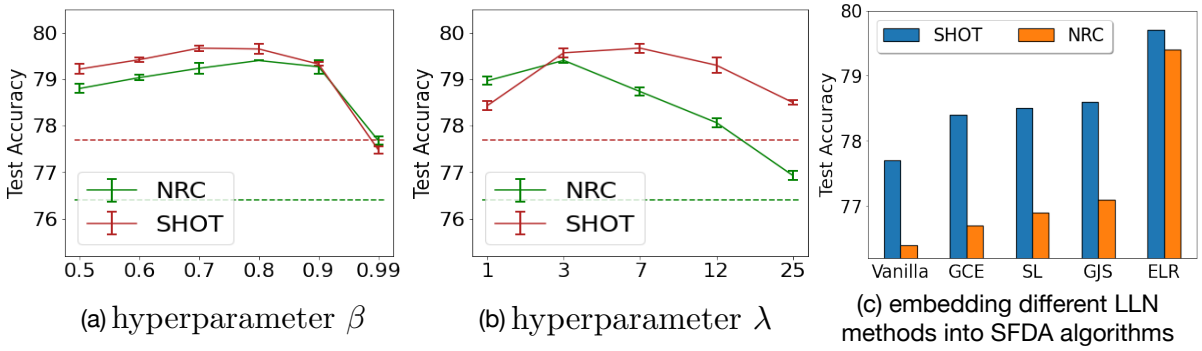
Analysis about hyperparameters β and λ. The hyperparameter β is chosen from {0.5, 0.6, 0.7, 0.8, 0.9, 0.99}, and λ is chosen from {1, 3, 7, 12, 25}. We conduct the sensitivity study on hyperparameters of ELR on the DomainNet dataset [96], which is shown in Figure 4.3(a-b). In each Figure, the study is conducted by fixing the other hyperparameter to the optimal one. The performance is robust to the hyperparameter β except $\beta = 0.99$. When $\beta = 0.99$, classifiers

Table 4.2: Accuracies (%) on Office-Home for ResNet50-based methods.

Method	SF	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	**Avg**
MCD [108]	✗	48.9	68.3	74.6	61.3	67.6	68.8	57.0	47.1	75.1	69.1	52.2	79.6	64.1
CDAN [79]	✗	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
SAFN [143]	✗	52.0	71.7	76.3	64.2	69.9	71.9	63.7	51.4	77.1	70.9	57.1	81.5	67.3
Symnets [159]	✗	47.7	72.9	78.5	64.2	71.3	74.2	64.2	48.8	79.5	74.5	52.6	82.7	67.6
MDD [158]	✗	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	**60.2**	82.3	68.1
TADA [133]	✗	53.1	72.3	77.2	59.1	71.2	72.1	59.7	53.1	78.4	72.4	60.0	82.9	67.6
BNM [20]	✗	52.3	73.9	80.0	63.3	72.9	74.9	61.7	49.5	79.7	70.5	53.6	82.2	67.9
BDG [146]	✗	51.5	73.4	78.7	65.3	71.5	73.7	65.1	49.7	81.1	74.6	55.1	84.8	68.7
SRDC [122]	✗	52.3	76.3	81.0	**69.5**	76.2	78.0	**68.7**	53.8	81.7	**76.3**	57.1	85.0	71.3
RSDA-MSTN [37]	✗	53.2	77.7	81.3	66.4	74.0	76.5	67.9	53.0	82.0	75.8	57.8	**85.4**	70.9
Source Only	✓	44.6	67.3	74.8	52.7	62.7	64.8	53.0	40.6	73.2	65.3	45.4	78.0	60.2
+ELR	✓	52.4	73.5	77.3	62.5	70.6	71.0	61.1	50.8	78.9	71.7	56.7	81.6	67.3
SHOT [71]	✓	57.1	78.1	81.5	68.0	78.2	78.1	67.4	54.9	82.2	73.3	58.8	84.3	71.8
+ELR	✓	**58.7**	**78.9**	**82.1**	68.5	79.0	77.5	68.2	57.1	81.9	74.2	59.5	84.9	**72.6**
G-SFDA [148]	✓	55.8	77.1	80.5	66.4	74.9	77.3	66.5	53.9	80.8	72.4	59.7	83.2	70.7
+ELR	✓	56.4	77.6	81.1	67.1	75.2	77.9	65.9	55.0	81.2	72.1	60.0	83.6	71.1
NRC [147]	✓	56.3	77.6	81.0	65.3	78.3	77.5	64.5	56.0	82.4	70.0	57.1	82.9	70.8
+ELR	✓	58.4	78.7	81.5	69.2	**79.5**	**79.3**	66.3	**58.0**	**82.6**	73.4	59.8	85.1	**72.6**

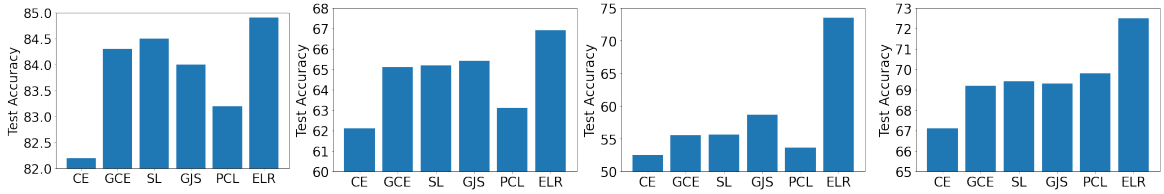
Table 4.3: Accuracies (%) on VisDA-C (Synthesis → Real) for ResNet101-based methods.

Method	SF	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Per-class
DANN [28]	✗	81.9	77.7	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
DAN [78]	✗	87.1	63.0	76.5	42.0	90.3	42.9	85.9	53.1	49.7	36.3	85.8	20.7	61.1
ADR [107]	✗	94.2	48.5	84.0	72.9	90.1	74.2	92.6	72.5	80.8	61.8	82.2	28.8	73.5
CDAN [79]	✗	85.2	66.9	83.0	50.8	84.2	74.9	88.1	74.5	83.4	76.0	81.9	38.0	73.9
SAFN [143]	✗	93.6	61.3	84.1	70.6	94.1	79.0	91.8	79.6	89.9	55.6	89.0	24.4	76.1
SWD [62]	✗	90.8	82.5	81.7	70.5	91.7	69.5	86.3	77.5	87.4	63.6	85.6	29.2	76.4
MDD [158]	✗	-	-	-	-	-	-	-	-	-	-	-	-	74.6
MCC [52]	✗	88.7	80.3	80.5	71.5	90.1	93.2	85.0	71.6	89.4	73.8	85.0	36.9	78.8
STAR [81]	✗	95.0	84.0	84.6	73.0	91.6	91.8	85.9	78.4	94.4	84.7	87.0	42.2	82.7
RWOT [144]	✗	95.1	80.3	83.7	**90.0**	92.4	68.0	92.5	82.2	87.9	78.4	**90.4**	**68.2**	84.0
Source Only	✓	60.9	21.6	50.9	67.6	65.8	6.3	82.2	23.2	57.3	30.6	84.6	8.0	46.6
+ELR	✓	95.4	45.7	89.7	69.8	94.1	97.1	**92.9**	80.1	89.7	52.8	83.3	4.3	74.6
SHOT [71]	✓	94.3	88.5	80.1	57.3	93.1	94.9	80.7	80.3	91.5	89.1	86.3	58.2	82.9
+ELR	✓	95.8	84.1	83.3	67.9	93.9	**97.6**	89.2	80.1	90.6	90.4	87.2	48.2	84.1
G-SFDA [148]	✓	96.0	87.6	85.3	72.8	95.9	94.7	88.4	79.0	92.7	93.9	87.2	43.7	84.8
+ELR	✓	**97.3**	89.1	**89.8**	79.2	**96.9**	97.5	92.2	**82.5**	**95.8**	**94.5**	87.3	34.5	**86.4**
NRC [147]	✓	96.9	89.7	84.0	59.8	95.9	96.6	86.5	80.9	92.8	92.6	90.2	60.2	85.4
+ELR	✓	97.1	**89.7**	82.7	62.0	96.2	97.0	87.6	81.2	93.7	94.1	90.2	58.6	85.8



(a) hyperparameter β (b) hyperparameter λ (c) embedding different LLN methods into SFDA algorithms

Figure 4.3: (a)-(b) show the test accuracy on the DomainNet dataset with respect to hyperparameters of ELR. (c) shows the test accuracy of incorporating various existing LLN methods into the SFDA methods on the DomainNet dataset.



(a) Office-31 (b) Office-Home (c) VisDA (d) DomainNet

Figure 4.4: Evaluation of label noise methods on SFDA problems. We use source models as an initialization of classifiers trained on target data and also use source models to annotate unlabeled target data. Then we treat the target datasets as noisy datasets and use different label noise methods to solve the memorization issue.

are sensitive to changes in learning curves. Thus, the performance degrades since the learning curves change quickly in the unbounded label noise scenarios. Meanwhile, the performance is also robust to the hyperparameter λ except when λ becomes too large. The hyperparameter λ is to balance the effects of existing SFDA algorithms and the effects of ELR. As we indicated in Tables 4.1-4.4, barely using ELR to address the SFDA problem is not comparable to these SFDA methods. Hence, a large value of λ makes neural networks neglect the effects of these SFDA methods, leading to degraded performance.

4.5.1 Discussion on Existing LLN Methods

As we formulate the SFDA as the problem of LLN, it is of interest to discuss some existing LLN methods. We mainly discuss existing LLN methods that can be easily embedded into the existing SFDA algorithms. Based on this principle, we choose GCE [161], SL [134] and GJS [24] that have been theoretically proved to be robust to symmetric and asymmetric label noise, which are bounded label noise. We highlight that a more recent method GJS [24] outperforms ELR in real-world noisy datasets. However, we will show that GJS is inferior to ELR in SFDA scenarios, because the underlying assumption for GJS does not hold in SFDA. Besides ELR, which leverages ETP, PCL is another method to leverage the same phenomenon, but we will

Table 4.4: Accuracies (%) on DomainNet for ResNet50-based methods.

Method	SF	R→C	R→P	R→S	C→R	C→P	C→S	P→R	P→C	P→S	S→R	S→C	S→P	**Avg**
MCD [108]	✗	61.9	69.3	56.2	79.7	56.6	53.6	83.3	58.3	60.9	81.7	56.2	66.7	65.4
DANN [28]	✗	63.4	73.6	72.6	86.5	65.7	70.6	86.9	73.2	70.2	85.7	75.2	70.0	74.5
DAN [78]	✗	64.3	70.6	58.4	79.4	56.7	60.0	84.5	61.6	62.2	79.7	65.0	62.0	67.0
COAL [120]	✗	73.9	75.4	70.5	89.6	70.0	71.3	89.8	68.0	70.5	88.0	73.2	70.5	75.9
MDD [158]	✗	77.6	75.7	**74.2**	89.5	74.2	**75.6**	90.2	76.0	**74.6**	86.7	72.9	73.2	78.4
Source Only	✓	53.7	71.6	52.9	70.8	49.5	58.3	85.2	59.6	59.1	30.6	74.8	65.7	61.0
+ELR	✓	70.2	81.7	61.7	79.9	63.8	67.0	90.0	72.1	66.8	85.1	78.5	68.8	73.8
SHOT [71]	✓	73.3	80.1	65.8	**91.4**	74.3	69.2	91.9	77.0	66.2	87.4	81.3	75.0	77.7
+ELR	✓	**78.0**	81.9	67.4	91.1	**75.9**	71.0	92.6	79.3	68.0	88.7	84.8	77.0	**79.7**
G-SFDA [148]	✓	65.8	78.9	60.2	80.5	64.7	64.6	89.3	69.9	63.6	86.4	78.8	71.1	72.8
+ELR	✓	69.4	80.9	60.6	81.3	67.2	66.4	90.2	73.2	64.9	87.6	82.1	71.0	74.6
NRC [147]	✓	69.8	81.1	62.9	83.4	74.4	66.3	90.3	73.4	65.2	88.2	82.2	75.8	76.4
+ELR	✓	75.6	**82.2**	65.7	91.2	77.2	68.5	**92.7**	**79.8**	67.5	**89.3**	**85.1**	**77.6**	79.4

show that it is also inappropriate for SFDA.

To show the effects of the existing LLN methods under the unbounded label noise, we test these LLN methods on various SFDA datasets with target data whose labels are generated by source models. As shown in Figure 4.4, GCE, SL, GJS, and PCL are better than CE but still not comparable to ELR. Our analysis indicates that ELR follows the principle of ETP, which is theoretically justified in SFDA scenarios by our Theorem 4.3.2. Methods GCE, SL, and GJS follow the bounded label noise assumption, which does not hold in SFDA. Hence, they perform worse than ELR in SFDA, even though GJS outperforms ELR in conventional LLN scenarios. PCL [160] utilizes ETP to purify noisy labels of target data, but it performs significantly worse than ELR. As the memorization speed of the unbounded label noise is very fast, and classifiers memorize noisy labels within a few iterations (shown in Figure 4.2), purifying noisy labels every epoch is inappropriate for SFDA. However, we notice that PCL has a relatively better performance in the DomainNet dataset [96] than in other datasets. The reason behind it is that the memorization speed in the DomainNet dataset is relatively slow than other datasets, which is shown in Figure 4.2. In conventional LLN scenarios, PCL does not suffer from the issue since the memorization speed is much lower than the conventional LLN scenarios.

In Figure 4.3(c), we also evaluate the performance by incorporating the existing LLN methods into the SFDA algorithms SHOT [71] and NRC [147]. Since PCL and SHOT assign pseudo labels to target data, PCL is incompatible with some existing SFDA methods and cannot be easily embedded into some SFDA algorithms. Hence, we only embed GCE, SL, GJS, and ELR into the SFDA algorithms. The figure illustrates that ELR still performs better than other LLN methods when incorporated into SHOT and NRC. We also notice that GCE, SL, and GJS provide marginal improvement to the vanilla SHOT and NRC methods. We think the label noise in SFDA datasets is the hybrid noise that consists of both bounded label noise and unbounded

label noise due to the non-linearity of neural networks. The GCE, SL, and GJS can address the bounded label noise, while ELR can address both bounded and unbounded label noise. Therefore, these experiments demonstrate that using ELR to leverage ETP can successfully address the unbounded label noise in SFDA.

4.6 Conclusion

We propose solving SFDA as the problem of LLN to address unbounded label noise in SFDA. We theoretically prove that the unbounded label noise exists as long as domain shift exists. We show that ETP exists in unbounded label noise, which can be leveraged to address the label noise. On the other hand, as a comparison, we also show that many existing LLN methods are unable to address unbounded label noise. Extensive experiments demonstrate that ETP can be exploited to improve the effects of many existing SFDA algorithms by ELR.

4.A Proofs for Theorem 4.3.1

Proof. The Bayes classifier f_S predicts $\mathbf{x}$ to the first component when

$$\log \frac{\Pr[y=1|X=\mathbf{x}]}{\Pr[y=-1|X=\mathbf{x}]} > 0. \tag{4.6}$$

Since the distributions of the two components with the same priors for the source domain are given by $\mathcal{N}(\boldsymbol{\mu_1}, \sigma^2\mathbf{I}_d)$ and $\mathcal{N}(\boldsymbol{\mu_2}, \sigma^2\mathbf{I}_d)$, respectively. Based on Bayes' rule, Eq. (4.6) is equivalent to

$$\log \frac{\Pr[X=\mathbf{x}|y=1]}{\Pr[X=\mathbf{x}|y=-1]} > 0 \tag{4.7}$$

Solving the left hand side of Eq. (4.7) by using the knowledge of two multivariate Gaussian distributions, we get

$$h_S(\mathbf{x}) \coloneqq \log \frac{\Pr[X=\mathbf{x}|y=1]}{\Pr[X=\mathbf{x}|y=-1]} = \frac{\mathbf{x}^\top(\boldsymbol{\mu_1}-\boldsymbol{\mu_2})}{\sigma^2} - \frac{\|\boldsymbol{\mu_1}\|_2^2 - \|\boldsymbol{\mu_2}\|_2^2}{2\sigma^2}. \tag{4.8}$$

So f_S predicts $\mathbf{x}$ to the first component when $h_S(\mathbf{x}) > 0$ and f_S predicts $\mathbf{x}$ to the second component when $h_S(\mathbf{x}) \leq 0$ The decision boundary is $\mathbf{z}$ such that $h_S(\mathbf{z}) = 0$. When there is no domain shift $\boldsymbol{\Delta} = \mathbf{0}$, we have $\mathcal{D}_S = \mathcal{D}_T$, and the mislabeling rate is the Bayes error, which is given by:

$$\Pr_{(\mathbf{x},y)\sim\mathcal{D}_S}[f_S(\mathbf{x}) \neq y] = \frac{1}{2}\Pr_{\mathbf{x}\sim\mathcal{N}(\boldsymbol{\mu_1},\sigma^2\mathbf{I}_d)}[h_S(\mathbf{x}) < 0|y=1] + \frac{1}{2}\Pr_{\mathbf{x}\sim\mathcal{N}(\boldsymbol{\mu_2},\sigma^2\mathbf{I}_d)}[h_S(\mathbf{x}) > 0|y=-1] \tag{4.9}$$

We first study the first term in Eq. (4.9):

$$\begin{aligned}
&\Pr_{\mathbf{x}\sim\mathcal{N}(\boldsymbol{\mu_1},\sigma^2\mathbf{I}_d)}[h_S(\mathbf{x}) < 0|y=1] \\
=& \int\cdots\int_{\{\mathbf{x}|\mathbf{x}^\top(\boldsymbol{\mu_1}-\boldsymbol{\mu_2})<\frac{\|\boldsymbol{\mu_1}\|_2^2-\|\boldsymbol{\mu_2}\|_2^2}{2}\}} \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}}\exp\left(-\frac{\|\mathbf{x}-\boldsymbol{\mu_1}\|_2^2}{2\sigma^2}\right)\mathrm{d}x_1\mathrm{d}x_2\cdots\mathrm{d}x_d \\
=& \int\cdots\int_{\{\mathbf{x}|-\infty<x_1,x_2,\ldots,x_{d-1}<\infty,d_0<x_d\}} \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}}\exp\left(-\frac{\sum_{i=1}^d x_i^2}{2\sigma^2}\right)\mathrm{d}x_1\mathrm{d}x_2\cdots\mathrm{d}x_d \\
=& \int_{d_0}^{\infty}\frac{1}{2\pi\sigma^2}\exp\left(-\frac{x_d^2}{2\sigma^2}\right)\mathrm{d}x_d \\
=& \Phi(-\frac{d_0}{\sigma}),
\end{aligned}$$

where the second equality is because of the rotationally symmetric property for isotropic Gaussian random vectors, Φ is the cumulative distribution function of the standard Gaussian distribution, and $d_0 = \|(\boldsymbol{\mu_2}-\boldsymbol{\mu_1})/2\|_2$. Applying the similar mathematical steps for the second term in Eq. (4.9), and take them into Eq. (4.9):

$$\Pr_{(\mathbf{x},y)\sim\mathcal{D}_S}[f_S(\mathbf{x}) \neq y] = \Phi(-\frac{\|\boldsymbol{\mu_2}-\boldsymbol{\mu_1}\|_2}{2\sigma}). \tag{4.10}$$

When there is no domain shift, the labeling error is the Bayes error, which is expressed by Eq. (4.10).

Then we consider the case when $\boldsymbol{\Delta} \neq \mathbf{0}$. The distributions of the first and the second component are $\mathcal{N}(\boldsymbol{\mu_1} + \boldsymbol{\Delta}, \sigma^2\mathbf{I}_d)$ and $\mathcal{N}(\boldsymbol{\mu_2} + \boldsymbol{\Delta}, \sigma^2\mathbf{I}_d)$, respectively. Notice that the decision boundary $\mathbf{z}$ is the affine hyperplane. Any shift paralleled to this affine hyperplane will not affect the final component predictions. The domain shift $\boldsymbol{\Delta}$ can be decomposed into the sum of two vectors: the one is paralleled to this affine hyperplane, and another is perpendicular to the hyperplane. It is straightforward to verify that $\boldsymbol{\mu_2} - \boldsymbol{\mu_1}$ is perpendicular to the hyperplane. Thus, we project the domain shift $\boldsymbol{\Delta}$ onto the vector $\boldsymbol{\mu_2} - \boldsymbol{\mu_1}$ to get the component of $\boldsymbol{\Delta}$ that is perpendicular to the hyperplane, which is given by:

$$\mathbf{c} = (\boldsymbol{\mu_2} - \boldsymbol{\mu_1})\frac{\boldsymbol{\Delta}^\top(\boldsymbol{\mu_2} - \boldsymbol{\mu_1})}{\|\boldsymbol{\mu_2} - \boldsymbol{\mu_1}\|_2^2}. \tag{4.11}$$

Since we assume $\boldsymbol{\Delta}$ is positively correlated to the vector $\boldsymbol{\mu_2} - \boldsymbol{\mu_1}$, $\alpha = \frac{\boldsymbol{\Delta}^\top(\boldsymbol{\mu_2}-\boldsymbol{\mu_1})}{\|\boldsymbol{\mu_2}-\boldsymbol{\mu_1}\|_2^2}$ can be regarded as the magnitude of the domain shift along the direction $\boldsymbol{\mu_2} - \boldsymbol{\mu_1}$. Note that the results also hold for the case where $\boldsymbol{\Delta}$ is negative correlated to $\boldsymbol{\mu_2} - \boldsymbol{\mu_1}$. The whole proof can be obtained by following the very similar proof steps for the positively correlated case.

The mislabeling rate of the optimal source classifier f_S on target data is:

$$\Pr_{(\mathbf{x},y)\sim\mathcal{D}_T}[f_S(\mathbf{x}) \neq y] = \frac{1}{2}\Pr_{\mathcal{N}(\boldsymbol{\mu_1}+\boldsymbol{\Delta},\sigma^2\mathbf{I}_d)}[h_S(\mathbf{x}) < 0|y = 1] + \frac{1}{2}\Pr_{\mathcal{N}(\boldsymbol{\mu_2}+\boldsymbol{\Delta},\sigma^2\mathbf{I}_d)}[h_S(\mathbf{x}) > 0|y = -1] \tag{4.12}$$

We first calculate the first term of Eq. (4.12). Following the same tricks discussed above:

$$\begin{aligned}
&\Pr_{\mathbf{x}\sim\mathcal{N}(\boldsymbol{\mu_1}+\boldsymbol{\Delta},\sigma^2\mathbf{I}_d)}[h_S(\mathbf{x}) < 0|y = 1] \\
=&\Pr_{\mathbf{x}\sim\mathcal{N}(\boldsymbol{\mu_1}+\mathbf{c},\sigma^2\mathbf{I}_d)}[h_S(\mathbf{x}) < 0|y = 1] \\
=&\int\cdots\int_{\{\mathbf{x}|\mathbf{x}^\top(\boldsymbol{\mu_1}-\boldsymbol{\mu_2})<\frac{\|\boldsymbol{\mu_1}\|_2^2-\|\boldsymbol{\mu_2}\|_2^2}{2}\}} \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}}\exp\left(-\frac{\|\mathbf{x}-\boldsymbol{\mu_1}-\boldsymbol{\Delta}\|_2^2}{2\sigma^2}\right)\mathrm{d}x_1\mathrm{d}x_2\cdots\mathrm{d}x_d \\
=&\int\cdots\int_{\{\mathbf{x}|-\infty<x_1,x_2,\ldots,x_{d-1}<\infty,d_1<x_d\}} \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}}\exp\left(-\frac{\sum_{i=1}^d x_i^2}{2\sigma^2}\right)\mathrm{d}x_1\mathrm{d}x_2\cdots\mathrm{d}x_d \\
=&\int_{d_1}^{\infty}\frac{1}{2\pi\sigma^2}\exp\left(-\frac{x_d^2}{2\sigma^2}\right)\mathrm{d}x_d \\
=&\Phi(-\frac{d_1}{\sigma}),
\end{aligned} \tag{4.13}$$

where $d_1 = \left\|\frac{\boldsymbol{\mu_2}-\boldsymbol{\mu_1}}{2} - \mathbf{c}\right\|_2 \operatorname{sign}(\left\|\frac{\boldsymbol{\mu_2}-\boldsymbol{\mu_1}}{2}\right\|_2 - \|\mathbf{c}\|_2)$.

Similarly, the second term is given by:

$$\Pr_{\mathbf{x}\sim\mathcal{N}(\boldsymbol{\mu_2}+\boldsymbol{\Delta},\sigma^2\mathbf{I}_d)}[h_S(\mathbf{x}) > 0|y = -1] =\Phi(-\frac{d_2}{\sigma}), \tag{4.14}$$

where $d_2 = \left\|\frac{\boldsymbol{\mu_2}-\boldsymbol{\mu_1}}{2} + \mathbf{c}\right\|_2$.

Taking Eq. (4.13) and Eq. (4.14) into Eq. (4.12), we have

$$\Pr_{(\mathbf{x},y)\sim\mathcal{D}_T}[f_S(\mathbf{x}) \neq y] = \frac{1}{2}\Phi(-\frac{d_1}{\sigma}) + \frac{1}{2}\Phi(-\frac{d_2}{\sigma}). \tag{4.15}$$

□

4.B Proofs for Theorem 4.3.2

Proof. Without loss of generality, we choose to assume $\boldsymbol{\mu_2} = \boldsymbol{\mu_1} + \sigma\mathbf{1}_d$ as the convenient way to present our results. From the proof for Theorem 4.3.1, we know that $\mathbf{x}_0 = \frac{\boldsymbol{\mu_1}+\boldsymbol{\mu_2}}{2} + \boldsymbol{\Delta}$ is at the decision boundary such that $h_T(\mathbf{x_0}) = 0$, where

$$h_T(\mathbf{x}) = \frac{\mathbf{x}^\top(\boldsymbol{\mu_1} - \boldsymbol{\mu_2})}{\sigma^2} - \frac{\|\boldsymbol{\mu_1} + \boldsymbol{\Delta}\|_2^2 - \|\boldsymbol{\mu_2} + \boldsymbol{\Delta}\|_2^2}{2\sigma^2}.$$

Let f_T be the optimal Bayes classifier for the target domain, which can be obtained the same way as f_S mentioned in 4.A. The equation $h_T(\mathbf{x_0}) = 0$ implies that

$$\Pr_{(\mathbf{x},y)\sim\mathcal{D}_T}[y = 1|X = \mathbf{x}_0] = \Pr_{(\mathbf{x},y)\sim\mathcal{D}_T}[y = -1|X = \mathbf{x}_0].$$

Note that $\mathbf{x}_0$ is on the affine hyperplane $\mathbf{z}$ where $h_T(\mathbf{z}) = 0$. Any data points on this hyperplane will have the equal probabilities to be correctly classified. We start from this hyperplane and calculate another point $\mathbf{x}_1$, where $\Pr_{(\mathbf{x},y)\sim\mathcal{D}_T}[y = 1|X = \mathbf{x}_1]$ is at least $99\Pr_{(\mathbf{x},y)\sim\mathcal{D}_T}[y = -1|X = \mathbf{x}_1]$. Thus, for any points that are mislabeled and far away from $\mathbf{x}_1$ will result in $\Pr_{(\mathbf{x},y)\sim\mathcal{D}_T}[y = 1|X = \mathbf{x}_1] \geq 0.99$. We first aim to find such a data point $\mathbf{x}_1$. Let $\mathbf{x}_1 = \mathbf{x}_0 - m_0\sigma\mathbf{1}_d$, where m_0 is the scalar measures the distance between the point $\mathbf{x}_1$ to the hyperplane $\mathbf{z}$. We need to find m_0 such that

$$\frac{P_T(\mathbf{x}_1|y = 1)}{P_T(\mathbf{x}_1|y = -1)} \geq 99, \tag{4.16}$$

where

$$\begin{aligned}
&\frac{P_T(\mathbf{x}_1|y = 1)}{P_T(\mathbf{x}_1|y = -1)} \\
&= \exp\Big(-\frac{\big\|\mathbf{x}_1 - \boldsymbol{\mu}_1 - \boldsymbol{\Delta}\big\|_2^2}{2\sigma^2} + \frac{\big\|\mathbf{x}_1 - \boldsymbol{\mu}_2 - \boldsymbol{\Delta}\big\|_2^2}{2\sigma^2}\Big) \\
&= \exp\Big(-\frac{\big\|\frac{\boldsymbol{\mu}_2-\boldsymbol{\mu}_1}{2} - m_0\sigma\mathbf{1}_d\big\|_2^2}{2\sigma^2} + \frac{\big\|\frac{\boldsymbol{\mu}_2-\boldsymbol{\mu}_1}{2} + m_0\sigma\mathbf{1}_d\big\|_2^2}{2\sigma^2}\Big) \\
&= \exp(m_0 d)
\end{aligned} \tag{4.17}$$

Taking Eq. (4.17) into Eq. (4.16), we get $m_0 \geq (\log 99)/d$. Since the isotropic Gaussian random vectors has the rotationally symmetric property, we can transform the integration of multivariate normal distribution to standard normal distribution with different intervals of integration. Then any data points from a region that have at most $\|\mathbf{x}_1 - \boldsymbol{\mu_1} - \boldsymbol{\Delta}\|_2$ distance to its

mean $\boldsymbol{\mu_1} + \boldsymbol{\Delta}$ will have at least 0.99 probability coming from the first component. Let the region $\mathbf{R}_1$ be:

$$\mathbf{R}_1 = \{\mathbf{x} : \left\|\mathbf{x} - \boldsymbol{\mu}_1 - \boldsymbol{\Delta}\right\|_2 \leq \|\mathbf{x}_1 - \boldsymbol{\mu_1} - \boldsymbol{\Delta}\|_2\}$$

Equivalently, taking $\mathbf{R}_1$ can be simplified:

$$\mathbf{R}_1 = \{\mathbf{x} : \left\|\mathbf{x} - \boldsymbol{\mu}_1 - \boldsymbol{\Delta}\right\|_2 \leq \sigma(\frac{\sqrt{d}}{2} - \frac{\log 99}{\sqrt{d}})\}$$

The region $\mathbf{R}_1$ is valid when data dimension d is large. This is realistic in practice. Since neural networks are usually dealing with high dimension data, for example $d \gg (1)$, the region $\mathbf{R}_1$ is valid.

On the other hand, we aim to find a region $\mathbf{R}_2$ where all data points are mislabeled. From the proof for Theorem 1, the source classifier h_S is given by

$$h_S(\mathbf{x}) = \frac{\mathbf{x}^\top(\boldsymbol{\mu_1} - \boldsymbol{\mu_2})}{\sigma^2} - \frac{\|\boldsymbol{\mu_1}\|_2^2 - \|\boldsymbol{\mu_2}\|_2^2}{2\sigma^2}. \tag{4.18}$$

Any data points are classified to the second component if $h_S(\mathbf{x}) < 0$. Hence

$$\mathbf{R}_2 = \{\mathbf{x} : \mathbf{x}^\top \mathbf{1}_d > \frac{\sigma d + 2\boldsymbol{\mu}_1^\top \mathbf{1}_d}{2}\}$$

We take the intersection of $\mathbf{R}_1$ and $\mathbf{R}_2$, all data points from this intersection are (1) having at least 0.99 probability coming from the first component, and (2) being classified to the second component. Formally, for $(\mathbf{x}, y) \sim \mathcal{D}_T$, if $\mathbf{x} \in \mathbf{R}_1 \bigcap \mathbf{R}_2$, then

$$\Pr[f_S(\mathbf{x}) \neq y] \geq 0.99, \tag{4.19}$$

We note that $\mathbf{x} \in \mathbf{R}_1 \bigcap \mathbf{R}_2$ is non-empty when $(\log 99)/d < \alpha$, where $\alpha = \frac{\boldsymbol{\Delta}^\top(\boldsymbol{\mu_2} - \boldsymbol{\mu_1})}{\|\boldsymbol{\mu_2} - \boldsymbol{\mu_1}\|_2^2}$ is the magnitude of the domain shift along with the direction $\boldsymbol{\mu_2} - \boldsymbol{\mu_1}$. Since $\mathbf{x}_1$ is chosen from $\mathbf{R}_1$, to verify that $\mathbf{R}_1 \bigcap \mathbf{R}_2$ is non-empty, we only need to verify that $\mathbf{x}_1$ also belongs to $\mathbf{R}_2$.

$\mathbf{x}_1 \in \mathbf{R}_2$ if and only if:

$$\begin{aligned}
\mathbf{x}_1^\top \mathbf{1}_d &> \frac{\sigma d + 2\boldsymbol{\mu}_1^\top \mathbf{1}_d}{2} \\
(\boldsymbol{\mu}_1 + \mathbf{c} + \frac{\sigma}{2}\mathbf{1}_d - m_0\sigma\mathbf{1}_d)^\top \mathbf{1}_d &> \frac{\sigma d + 2\boldsymbol{\mu}_1^\top \mathbf{1}_d}{2} \\
(\boldsymbol{\mu}_1 + \alpha\sigma\mathbf{1}_d + \frac{\sigma}{2}\mathbf{1}_d - m_0\sigma\mathbf{1}_d)^\top \mathbf{1}_d &> \frac{\sigma d + 2\boldsymbol{\mu}_1^\top \mathbf{1}_d}{2} \\
(\alpha - m_0)\sigma d &> 0,
\end{aligned}$$

where $\mathbf{c} = (\boldsymbol{\mu_2} - \boldsymbol{\mu_1})\frac{\boldsymbol{\Delta}^\top(\boldsymbol{\mu_2} - \boldsymbol{\mu_1})}{\|\boldsymbol{\mu_2} - \boldsymbol{\mu_1}\|_2^2}$.

Therefore, if $\alpha > m_0 \geq (\log 99)/d$, $\mathbf{R}_1 \bigcap \mathbf{R}_2$ is non-empty.

Next, we show $\Pr_{(\mathbf{x},y)\sim\mathcal{D}_T}[\mathbf{x} \in \mathbf{R}]$ increases as α increases.

Let event $\mathbf{A}_0$ be a set of $\mathbf{x}$ such that they are mislabeled by f_S (i.e. $f_S(\mathbf{x}) \neq y$). Let event $\mathbf{A}_1$ be a set of $\mathbf{x}$ such that they are from the first component but are mislabeled to the second

component with a probability $\Pr[f_S(\mathbf{x} \neq y)] < 0.99$. Let event $\mathbf{A}_2$ be a set of $\mathbf{x}$ such that they are from the second component but are mislabeled to the first component with a probability $\Pr[f_S(\mathbf{x} \neq y)] < 0.99$. Thus

$$\Pr_{(\mathbf{x},y)\sim\mathcal{D}_T}[\mathbf{x} \in \mathbf{R}] = \Pr_{(\mathbf{x},y)\sim\mathcal{D}_T}[\mathbf{A}_0] - \Pr_{(\mathbf{x},y)\sim\mathcal{D}_T}[\mathbf{A}_1] - \Pr_{(\mathbf{x},y)\sim\mathcal{D}_T}[\mathbf{A}_2] \tag{4.20}$$

Let event $\mathbf{A}_3$ be a set of $\mathbf{x}$ such that they are from the first component such that $\Pr[f_S(\mathbf{x} \neq y)] < 0.99$ or $\Pr[f_S(\mathbf{x} = y)] < 0.99$. Let event $\mathbf{A}_4$ be a set of $\mathbf{x}$ such that they are from the second component but are mislabeled to the first component. For $\Pr[\mathbf{A}_3]$,

$$\Pr_{(\mathbf{x},y)\sim\mathcal{N}(\boldsymbol{\mu_1}+\boldsymbol{\Delta},\sigma^2\mathbf{I}_d)}[\mathbf{A}_3] = \Pr_{(\mathbf{x},y)\sim\mathcal{N}(\boldsymbol{\mu_1}+\boldsymbol{\Delta},\sigma^2\mathbf{I}_d)}[\mathbf{R}_1^{\complement}],$$

which does not change as the domain shift $\boldsymbol{\Delta}$ varies. Meanwhile,

$$\Pr_{(\mathbf{x},y)\sim\mathcal{N}(\boldsymbol{\mu_2}+\boldsymbol{\Delta},\sigma^2\mathbf{I}_d)}[\mathbf{A}_4] = \Phi(-\frac{\left\|\frac{\boldsymbol{\mu_2}-\boldsymbol{\mu_1}}{2} + \mathbf{c}\right\|_2}{\sigma}),$$

which is given by Eq. (4.14). By our assumption, the domain shift $\boldsymbol{\Delta}$ is positively correlated with the vector $\boldsymbol{\mu_2} - \boldsymbol{\mu_1}$. So when α increases, $\Pr_{(\mathbf{x},y)\sim\mathcal{N}(\boldsymbol{\mu_2}+\boldsymbol{\Delta},\sigma^2\mathbf{I}_d)}[\mathbf{A}_4]$ decreases.

Since $\mathbf{A}_1 \subseteq \mathbf{A}_3$ and $\mathbf{A}_2\mathbf{A}_4$, the probability measure on $\mathbf{R}$ is given by:

$$\begin{aligned}\Pr_{(\mathbf{x},y)\sim\mathcal{D}_T}[\mathbf{x} \in \mathbf{R}] &= \Pr_{(\mathbf{x},y)\sim\mathcal{D}_T}[\mathbf{A}_0] - \Pr_{(\mathbf{x},y)\sim\mathcal{D}_T}[\mathbf{A}_1] - \Pr_{(\mathbf{x},y)\sim\mathcal{D}_T}[\mathbf{A}_2] \\ &\geq \Pr_{(\mathbf{x},y)\sim\mathcal{D}_T}[\mathbf{A}_0] - \Pr_{(\mathbf{x},y)\sim\mathcal{D}_T}[\mathbf{A}_3] - \Pr_{(\mathbf{x},y)\sim\mathcal{D}_T}[\mathbf{A}_4],\end{aligned} \tag{4.21}$$

where the first term is the mislabeling rate that increases as α increases (given by Theorem 4.3.1); the second term is a constant; the third term decreases as as α increases. The equality in Eq. (4.21) holds when $\alpha \to \infty$. Therefore, when the magnitude of the domain shift α increases, the lower bound of $\Pr_{(\mathbf{x},y)\sim\mathcal{D}_T}[\mathbf{x} \in \mathbf{R}]$ increases, which forces more points to break the conventional LLN assumption.

□

4.C Proofs for Lemma 4.3.3

Proof. We first introduce the background of noise-robust loss functions. As indicated in [85], the loss function ℓ is defined to be noise robust if $\sum_{j=1}^{K} \ell(h(\mathbf{x}), j) = C$, where C is a positive constant. Existing noise robust loss functions such as mean absolute error (MAE) [89], reverse cross entropy (RCE) [134], normalized cross entropy (NCE) [85], and normalized focal loss (NFL) satisfy this condition. Note that generalized cross entropy (GCE [161]) extends MAE and symmetric loss (SL [134]) extends RCE. So we study GCE and SL in our experiments instead studying MAE and RCE. Another noise robust loss function GJS [24] is shown to be tightly bounded around $\sum_{j=1}^{K} \ell(h(\mathbf{x}), j)$. All these methods have shown to be noise tolerant under either bounded random label noise or bounded class-conditional label noise with additional assumption that $R(h^\star) = 0$. We show that under the same assumption with unbounded label noise datasets, these methods are not noise tolerant.

Let $\eta_{yk}(\mathbf{x})$ be the $\Pr[\tilde{Y} = k|Y = y, X = \mathbf{x}]$ probability of observing a noisy label k given the ground-truth label y and a sample $\mathbf{x}$. Let $\eta_y(\mathbf{x}) = \sum_{k\neq y} \eta_{yk}(\mathbf{x})$. The risk of h under noisy data is given by

$$\begin{aligned}
\widetilde{R}(h) =& \mathbb{E}_{\mathbf{x},\tilde{y}}[\ell_{\text{LLN}}(h(\mathbf{x}), \tilde{y})] \\
=& \mathbb{E}_{\mathbf{x}}\mathbb{E}_{y|\mathbf{x}}\mathbb{E}_{\tilde{y}|\mathbf{x},y}[\ell_{\text{LLN}}(h(\mathbf{x}), \tilde{y})] \\
=& \mathbb{E}_{\mathbf{x},y}\Big[(1 - \eta_y(\mathbf{x}))\ell_{\text{LLN}}(h(\mathbf{x}), y) + \sum_{k\neq y} \eta_{yk}(\mathbf{x})\ell_{\text{LLN}}(h(\mathbf{x}), k)\Big] \\
=& \mathbb{E}_{\mathbf{x},y}\Big[(1 - \eta_y(\mathbf{x}))(\sum_{k=1}^{K} \ell_{\text{LLN}}(h(\mathbf{x}), k) - \sum_{k\neq y} \ell_{\text{LLN}}(h(\mathbf{x}), k)) + \sum_{k\neq y} \eta_{yk}(\mathbf{x})\ell_{\text{LLN}}(h(\mathbf{x}), k)\Big] \\
=& \mathbb{E}_{\mathbf{x},y}\Big[(1 - \eta_y(\mathbf{x}))(C - \sum_{k\neq y} \ell_{\text{LLN}}(h(\mathbf{x}), k)) + \sum_{k\neq y} \eta_{yk}(\mathbf{x})\ell_{\text{LLN}}(h(\mathbf{x}), k)\Big] \\
=& \mathbb{E}_{\mathbf{x},y}\Big[(1 - \eta_y(\mathbf{x}))C\Big] - \mathbb{E}_{\mathbf{x},y}\Big[\sum_{k\neq y} (1 - \eta_y(\mathbf{x}) - \eta_{yk}(\mathbf{x}))\ell_{\text{LLN}}(h(\mathbf{x}), k)\Big]. \qquad (4.22)
\end{aligned}$$

Since Eq. (4.22) holds for both $\tilde{h}^\star$ and $h^\star$, we have

$$\widetilde{R}(\tilde{h}^\star) = \mathbb{E}_{\mathbf{x},y}\Big[(1 - \eta_y(\mathbf{x}))C\Big] - \mathbb{E}_{\mathbf{x},y}\Big[\sum_{k\neq y} (1 - \eta_y(\mathbf{x}) - \eta_{yk}(\mathbf{x}))\ell_{\text{LLN}}(\tilde{h}^\star(\mathbf{x}), k)\Big] \qquad (4.23)$$

and

$$\widetilde{R}(h^\star) = \mathbb{E}_{\mathbf{x},y}\Big[(1 - \eta_y(\mathbf{x}))C\Big] - \mathbb{E}_{\mathbf{x},y}\Big[\sum_{k\neq y} (1 - \eta_y(\mathbf{x}) - \eta_{yk}(\mathbf{x}))\ell_{\text{LLN}}(h^\star(\mathbf{x}), k)\Big]. \qquad (4.24)$$

As $\tilde{h}^\star$ is the minimizer of $\widetilde{R}(h)$, $\widetilde{R}(\tilde{h}^\star) \leq \widetilde{R}(h^\star)$. Then we combine Eq. (4.23) and Eq. (4.24), we have

$$\mathbb{E}_{\mathbf{x},y}\Big[\sum_{k\neq y} (1 - \eta_y(\mathbf{x}) - \eta_{yk}(\mathbf{x}))(\ell_{\text{LLN}}(h^\star(\mathbf{x}), k) - \ell_{\text{LLN}}(\tilde{h}^\star(\mathbf{x}), k))\Big] \leq 0. \qquad (4.25)$$

We note that $\ell_{\text{LLN}}(\tilde{h}^\star(\mathbf{x}), k) \geq \ell_{\text{LLN}}(h^\star(\mathbf{x}), k)$ implies $p_k(\mathbf{x}) = 0$ and $p_y(\mathbf{x}) = 1$ for $k \neq y$, where $p_k(\mathbf{x})$ is the probability output by $\tilde{h}^\star$ for predicting the sample $\mathbf{x}$ to be the class k. This argument is proved given by [134, 89, 148, 85] (Theorem 1&2 in [89], Theorem 1 in [134], Lemma 1&2 in [85] and Theorem 1&2 in [24]).

To let $\ell_{\text{LLN}}(\tilde{h}^\star(\mathbf{x}), k) \geq \ell_{\text{LLN}}(h^\star(\mathbf{x}), k)$ holds for all inputs $\mathbf{x}$, previous studies assume the bounded label noise, which is given by

$$1 - \eta_y(\mathbf{x}) - \eta_{yk}(\mathbf{x}) > 0 \;\; \forall \mathbf{x} \;\text{ s.t. }\; P(X = \mathbf{x}) > 0. \qquad (4.26)$$

For random label noise which assumes that the mislabeling probability from the ground-truth label to any other label is the same for all inputs, i.e. $\eta_{ji}(\mathbf{x}) = a_0 \;\forall i \neq j$, where a_0 is a

constant. Let $\eta = (K-1)a_0$, then Eq. (4.26) is degraded to

$$\begin{aligned} 1 - \eta - \frac{\eta}{K-1} &> 0 \\ 1 &> \frac{K}{K-1}\eta \\ \eta &< 1 - \frac{1}{K}. \end{aligned}$$

This bounded assumption is commonly assumed by [134, 89, 148, 85] (Theorem 1 in [89], Theorem 1 in [134], Lemma 1 in [85] and Theorem 1 in [24]).

For class-conditional label noise, which assumes the $\eta_{ji}(\mathbf{x}_1) = \eta_{ji}(\mathbf{x}_2)$ for any inputs $\mathbf{x}_1$ and $\mathbf{x}_2$. Let $\eta_{ji}(\mathbf{x}) = \eta_{ji}$, Then the bounded assumption Eq. (4.26) is degraded to

$$\eta_{yk} < 1 - \eta_y.$$

This bounded assumption is also commonly assumed, and it can be found in Theorem 2 in [89], Theorem 1 in [134], 2 in [85] and Theorem 2 in [24].

However, in SFDA, we proved that there exists $\mathbf{R} \subset \mathcal{X}$ and for $\mathbf{x} \in \mathbf{R}$, $\Pr[\tilde{y} \neq y | \mathbf{x} \in \mathbf{R}] = 1$. As the label noise is unbounded for $\mathbf{x} \in \mathbf{R}$,

$$1 - \eta_y(\mathbf{x}) - \eta_{yk}(\mathbf{x}) = 1 - 1 - \eta_{yk}(\mathbf{x}) < 0 \ \forall \mathbf{x} \in \mathbf{R}. \tag{4.27}$$

Given the result in Eq. (4.27), and combined it with the Eq. (4.25), we have

$$\ell_{\text{LLN}}(\tilde{h}^\star(\mathbf{x}), k) \leq \ell_{\text{LLN}}(h^\star(\mathbf{x}), k).$$

Note that only $\ell_{\text{LLN}}(\tilde{h}^\star(\mathbf{x}), k) \geq \ell_{\text{LLN}}(h^\star(\mathbf{x}), k)$ means $p_k(\mathbf{x}) = 0$ for $k \neq y$ and $p_y(\mathbf{x}) = 1$ for $k \neq y$. It means that the optimal classifier $\tilde{h}^\star$ from noisy data can make correct predictions on any inputs, which is consistent with the optimal classifier $h^\star$ obtained from clean data.

As for the condition $\ell_{\text{LLN}}(\tilde{h}^\star(\mathbf{x}), k) \leq \ell_{\text{LLN}}(h^\star(\mathbf{x}), k)$, we can get $p_k(\mathbf{x}) = 1$ for a $k \neq y$, which means that the optimal classifier $\tilde{h}^\star$ from noisy data cannot make correct predictions on samples $\mathbf{x} \in \mathbf{R}$. To verify this, we use the robust loss function RCE ℓ_{RCE} as an example, and it can be easily generalized to other robust los functions mentioned above. Based on the definition of the RCE loss [134], we have

$$\begin{aligned} \ell_{\text{RCE}}(\tilde{h}^\star(\mathbf{x}), k) =& C_{\text{RCE}}(1 - p_k(\mathbf{x})) \\ \ell_{\text{RCE}}(h^\star(\mathbf{x}), k) =& C_{\text{RCE}}, \end{aligned}$$

where $C_{\text{RCE}} > 0$ is a constant. The above equations show that any $0 \leq p_k(\mathbf{x}) \leq 1$ can make the condition $\ell_{\text{LLN}}(\tilde{h}^\star(\mathbf{x}), k) \leq \ell_{\text{LLN}}(h^\star(\mathbf{x}), k)$ hold. Meanwhile, $\tilde{h}^\star$ is the global minimizer of the risk over the noisy data, which makes $\tilde{h}^\star$ memorize the noisy dataset.

Therefore, $\tilde{h}^\star$ makes incorrect predictions for $\mathbf{x} \in \mathbf{R}$ such that $p_k(\mathbf{x}) = 1$ for a $k \neq y$, and $h^\star$ is the global optimal over clean data, which gives correct predictions for $\mathbf{x} \in \mathbf{R}$ such that $p_k(\mathbf{x}) = 1$ for a $k = y$. That completes the proof as $h^\star$ makes different predictions on $\mathbf{x} \in \mathbf{R}$ compared to $\tilde{h}^\star$.

□

4.D Proofs for Theorem 4.4.1

The proof for Theorem 4.4.1 is partially adopted from [75]. Note that we are dealing with unbounded label noise, whereas the bounded label noise is considered in [75]. As indicated in [75], T is set as the smallest positive integer such that $\theta_T^\top \boldsymbol{\mu} \geq 0.1$, and $T = \Omega(1/\eta)$ with high probability. Parameters θ is initialized by Kaiming initialization [42] that $\theta_0 \sim \mathcal{N}(0, \frac{2}{d}\mathbf{I}_d)$, and $|\theta_0^\top \boldsymbol{\mu}|$ converges in probability to 0. For simplicity, we assume $\theta_0 = 0$ without loss of generality. The proof consists of two parts. The first part is to show that θ_{T-1} is highly positively correlated with the ground truth classifier. The second part is to show that the prediction accuracy on mislabeled samples can be represented as the correlation between the learned classifier and the ground truth classifier.

Proof. **To begin with, we show the first part.** Let samples $\mathbf{x}_i = y_i(\boldsymbol{\mu} - \sigma \mathbf{z}_i)$, where $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I}_d)$. The gradient of the logistic loss function with respect to the parameter θ is given by:

$$\begin{aligned}\nabla_\theta \mathcal{L}(\theta_t) =& \frac{1}{2n}\sum_{i=1}^{n} \mathbf{x}_i(\tanh(\theta_t^\top \mathbf{x}_i) - \tilde{y}_i) \\ =& \underbrace{-\frac{1}{2n}\sum_{i=1}^{n} \tilde{y}_i \mathbf{x}_i}_{①} + \underbrace{\frac{1}{2n}\sum_{i=1}^{n} \mathbf{x}_i \tanh(\theta_t^\top \mathbf{x}_i)}\end{aligned} \tag{4.28}$$

Then we will show that $-\boldsymbol{\mu}^\top \nabla_\theta \mathcal{L}(\theta_t)$ is lower bounded by a positive number. We first show the bound on ①in Eq. (4.28). Since $\mathbf{x}_i$ is sampled from standard normal distribution, $\frac{1}{n}\sum_{i=1}^{n} \tilde{y}_i \boldsymbol{\mu}^\top \mathbf{x}_i$ has limited variance. By the law of large number, $\frac{1}{n}\sum_{i=1}^{n} \tilde{y}_i \boldsymbol{\mu}^\top \mathbf{x}_i$ converges in probability to its mean. Therefore,

$$\begin{aligned}\mathbb{E}[\tilde{y}\mathbf{x}^\top \boldsymbol{\mu}] =& \mathbb{E}[\tilde{y}\boldsymbol{\mu}^\top \mathbf{x} \mathbb{1}\{y\mathbf{x}^\top \boldsymbol{\mu} \leq r\}] + \mathbb{E}[\tilde{y}\boldsymbol{\mu}^\top \mathbf{x} \mathbb{1}\{y\mathbf{x}^\top \boldsymbol{\mu} > r\}] \\ =& \mathbb{E}[\mathbb{E}[\tilde{y}\boldsymbol{\mu}^\top \mathbf{x} \mathbb{1}\{y\mathbf{x}^\top \boldsymbol{\mu} \leq r\}]|y] \\ &+ \mathbb{E}[\mathbb{E}[\tilde{y}\boldsymbol{\mu}^\top \mathbf{x} \mathbb{1}\{y\mathbf{x}^\top \boldsymbol{\mu} > r\}]|y] \\ =& \mathbb{E}[-\boldsymbol{\mu}^\top \mathbf{x} \mathbb{1}\{\mathbf{x}^\top \boldsymbol{\mu} \leq r\}|y = 1] + \mathbb{E}[\boldsymbol{\mu}^\top \mathbf{x} \mathbb{1}\{\mathbf{x}^\top \boldsymbol{\mu} > r\}|y = 1]\end{aligned}$$

Note that $\mathbf{x}|y = 1$ is a Gaussian random vector with independent entries, we have $\mathbf{x}^\top \boldsymbol{\mu} \overset{\mathrm{d}}{=} w + 1$, where $w \sim \mathcal{N}(0, \sigma^2)$. Therefore, the above expectation is equivalent to

$$\begin{aligned}\mathbb{E}[\tilde{y}\mathbf{x}^\top \boldsymbol{\mu}] =& -\int_{-\infty}^{r-1} (w+1)\, \mathrm{d}\mathbb{P}_w + \int_{r-1}^{\infty} (w+1)\, \mathrm{d}\mathbb{P}_w \\ =& -\int_{-\infty}^{r-1} w\, \mathrm{d}\mathbb{P}_w + \int_{r-1}^{+\infty} w\, \mathrm{d}\mathbb{P}_w - \int_{-\infty}^{r-1} \mathrm{d}\mathbb{P}_w + \int_{r-1}^{+\infty} \mathrm{d}\mathbb{P}_w \\ =& \int_{r-1}^{1-r} \mathrm{d}\mathbb{P}_w - \int_{-\infty}^{r-1} w\, \mathrm{d}\mathbb{P}_w + \int_{r-1}^{+\infty} w\, \mathrm{d}\mathbb{P}_w \\ =& \mathrm{Erf}[\frac{1-r}{\sqrt{2}\sigma}] + 2\frac{\sigma}{\sqrt{2\pi}} \exp\big(-\frac{(r-1)^2}{2\sigma^2}\big),\end{aligned} \tag{4.29}$$

where $\mathrm{Erf}[x] = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} \mathrm{d}t$. Note that $r < 1$, which means that most half of samples are mislabeled. Thus

$$\frac{1}{2}\mathbb{E}[\tilde{y}_i \boldsymbol{\mu}^\top \mathbf{x}_i] = \frac{1}{2}\mathrm{Erf}[\frac{1-r}{\sqrt{2}\sigma}] + \frac{\sigma}{\sqrt{2\pi}} \exp\big(-\frac{(r-1)^2}{2\sigma^2}\big) > 0.$$

Now we deal with the in in Eq. (4.28).

$$\begin{aligned}\frac{1}{2n}|\boldsymbol{\mu}^\top(\sum_{i=1}^{n} \tanh(\theta_t^\top x_i))| =& \frac{1}{2n}|\mathbf{q}^\top \mathbf{p}| \\ \leq& \frac{1}{2n} \|\mathbf{q}\|_2 \|\mathbf{p}\|_2 ,\end{aligned} \tag{4.30}$$

$\mathbf{q} = (\boldsymbol{\mu}^\top \mathbf{x}_1, \boldsymbol{\mu}^\top \mathbf{x}_2, \ldots, \boldsymbol{\mu}^\top \mathbf{x}_n) \in \mathbb{R}^n$, and $\mathbf{p} = (\tanh(\theta_t^\top x_1), \tanh(\theta_t^\top x_2), \ldots, \tanh(\theta_t^\top x_n)) \in \mathbb{R}^n$.

By triangle inequality of the norm,

$$\|\mathbf{q}\|_2 = \|\mathbf{q} - \mathbf{1} + \mathbf{1}\|_2 \leq \|\mathbf{q} - \mathbf{1}\|_2 + \|\mathbf{1}\|_2 = \sqrt{n} + \|\mathbf{q} - \mathbf{1}\|_2 ,$$

where $\mathbf{q} - \mathbf{1}$ is a random vector with Gaussian coordinates. By Lemma 4.D.1,

$$\|\mathbf{q} - \mathbf{1}\|_2 / \sigma \leq 2\sigma \sqrt{n} \tag{4.31}$$

with probability $1 - \delta$ when $n \geq c_1 \log 1/\delta$, where c_1 is a constant.

On the other hand,

$$\begin{aligned}\left\|\mathbf{p} - \tanh(\theta_t^\top \boldsymbol{\mu})\mathbf{1}_n + \tanh(\theta_t^\top \boldsymbol{\mu})\mathbf{1}_n\right\|_2 \leq& \left\|\tanh(\theta_t^\top \boldsymbol{\mu})\mathbf{1}_n\right\|_2 + \left\|\mathbf{p} - \tanh(\theta_t^\top \boldsymbol{\mu})\mathbf{1}_n\right\|_2 \\ \leq& \left\|\tanh(\theta_t^\top \boldsymbol{\mu})\mathbf{1}_n\right\|_2 + \|\theta_t\|_2 \|\mathbf{q} - 1\|_2 \\ =& \tanh(\theta_t^\top \boldsymbol{\mu}) \sqrt{n} + 2\sigma \sqrt{n} \|\theta_t\|_2 ,\end{aligned} \tag{4.32}$$

where the second inequality is by Lemma 9 from [75], the last inequality by Lemma 4.D.1.

Then we take Eq. (4.30) and Eq.(4.32) together, and then take them and Eq.(4.29) into $-\boldsymbol{\mu}^\top \nabla_\theta \mathcal{L}(\theta_t)$, which gives us:

$$-\nabla_\theta \mathcal{L}(\theta_t)^\top \boldsymbol{\mu} \geq \frac{1}{2}\mathrm{Erf}[\frac{1-r}{\sqrt{2}\sigma}] + \frac{\sigma}{\sqrt{2\pi}} \exp\big(-\frac{(r-1)^2}{2\sigma^2}\big) - \sigma(\tanh(\theta_t^\top \boldsymbol{\mu}) + 2\sigma \|\theta_t\|_2) \tag{4.33}$$

By Lemma 8 from [75], we have $\sup_{\theta \in \mathbb{R}^d} \|\nabla_\theta \mathcal{L}(\theta)\|_2 \leq 1 + 2\sigma$. Therefore, Eq. (4.33) can be rewritten as:

$$\begin{aligned}\frac{-\nabla_\theta \mathcal{L}(\theta_t)^\top \boldsymbol{\mu}}{\|\nabla_\theta \mathcal{L}(\theta_t)\|_2} \geq& \frac{\mathrm{Erf}[\frac{1-r}{\sqrt{2}\sigma}] + 2\frac{\sigma}{\sqrt{2\pi}} \exp\big(-\frac{(r-1)^2}{2\sigma^2}\big)}{1 + 2\sigma} - \frac{\sigma(\tanh(\theta_t^\top \boldsymbol{\mu}) + 2\sigma \|\theta_t\|_2)}{1 + 2\sigma} \\ \geq& \frac{b_0}{1 + 2\sigma} - \frac{\sigma(\tanh(\theta_t^\top \boldsymbol{\mu}) + 2\sigma \|\theta_t\|_2)}{1 + 2\sigma},\end{aligned} \tag{4.34}$$

where we let $b_0 = \frac{1}{2}\mathrm{Erf}[\frac{1-r}{\sqrt{2}\sigma}] + \frac{\sigma}{\sqrt{2\pi}} \exp\big(-\frac{(r-1)^2}{2\sigma^2}\big)$.

Then we prove $\frac{-\nabla_\theta \mathcal{L}(\theta_t)^\top \boldsymbol{\mu}}{\|\nabla_\theta \mathcal{L}(\theta_t)\|_2} \geq \frac{1}{10}\frac{b_0}{1+2\sigma}$ by mathematical induction, which can help us get rid of the dependence on θ_t for the lower bound in Eq. (4.34).

For $t = 0$, the inequality holds trivially. By the gradient descent algorithm, $\theta_{t+1} = -\eta \sum_{i=0}^{t} \nabla_\theta \mathcal{L}(\theta_i)$, where $-\boldsymbol{\mu}^\top \nabla_\theta \mathcal{L}(\theta_i) / \|\nabla_\theta \mathcal{L}(\theta_i)\|_2 \geq \frac{1}{10}\frac{b_0}{1+2\sigma}$.

$$\begin{aligned}\frac{\theta_{t+1}^\top \boldsymbol{\mu}}{\|\theta_{t+1}\|_2} \geq & \frac{-\eta \sum_{i=0}^{t} \boldsymbol{\mu}^\top \nabla_\theta \mathcal{L}(\theta_i)}{\eta \left\| \sum_{i=0}^{t} \nabla_\theta \mathcal{L}(\theta_i) \right\|_2} \\ \geq & \frac{\frac{1}{10}\frac{b_0}{1+2\sigma}(\sum_{i=0}^{t} \|\nabla_\theta \mathcal{L}(\theta_i)\|_2)}{\sum_{i=0}^{t} \|\nabla_\theta \mathcal{L}(\theta_i)\|_2} \\ \geq & \frac{1}{10}\frac{b_0}{1+2\sigma}\end{aligned}$$

As $t + 1 < T$, we have $\|\theta_{t+1}\|_2 \leq 10\frac{1+2\sigma}{b_0}\theta_{t+1}^\top \boldsymbol{\mu} \leq \frac{1+2\sigma}{b_0}$. Taking it into Eq. (4.34), we have

$$\frac{-\nabla_\theta \mathcal{L}(\theta_t)^\top \boldsymbol{\mu}}{\|\nabla_\theta \mathcal{L}(\theta_t)\|_2} \geq \frac{b_0}{1+2\sigma} - \frac{\sigma(0.1 + \frac{1+2\sigma}{b_0})}{1+2\sigma}$$

To show $\frac{-\nabla_\theta \mathcal{L}(\theta_t)^\top \boldsymbol{\mu}}{\|\nabla_\theta \mathcal{L}(\theta_t)\|_2}$ is lower bounded by $\frac{1}{10}\frac{b_0}{1+2\sigma}$, we need to have

$$h(\sigma) = \frac{9}{10}\frac{b_0}{1+2\sigma} - \sigma(0.1 + \frac{1+2\sigma}{b_0}) > 0$$

It is straightforward to verify that $h(\sigma = 0) > 0$ and it can be verified that when $0 < \sigma < c_0$, we have $h'(\sigma) > 0$. Therefore, for $0 < \sigma < c_0$ and any $t < T - 1$

$$\frac{-\nabla_\theta \mathcal{L}(\theta_t)^\top \boldsymbol{\mu}}{\|\nabla_\theta \mathcal{L}(\theta_t)\|_2} \geq \frac{1}{10}\frac{b_0}{1+2\sigma}$$

Hence by gradient descent algorithm $\theta_T = -\eta \sum_{i=0}^{T-1} \nabla_\theta \mathcal{L}(\theta_i)$ and the same proof above, we have

$$\frac{\theta_T^\top \boldsymbol{\mu}}{\|\theta_T\|_2} \geq \frac{1}{10}\frac{b_0}{1+2\sigma} \tag{4.35}$$

For the second part: the prediction accuracy on mislabeled sample set B converges in probability to its mean. Therefore, the expectation of the prediction accuracy on mislabeled samples is given by

$$\begin{aligned}\mathbb{E}[\mathbb{1}\{\text{sign}(\theta_T^\top \mathbf{x}) = y\}] = & \mathbb{E}[\mathbb{1}\{\text{sign}(y\theta_T^\top(\boldsymbol{\mu} - \sigma\mathbf{z})) = y\}] \\ = & \mathbb{E}[\mathbb{1}\{\text{sign}(\theta_T^\top(\boldsymbol{\mu} - \sigma\mathbf{z})) = 1\}] \\ = & \Pr[\sigma\theta_T^\top \mathbf{z} > \theta_T^\top \boldsymbol{\mu}]\end{aligned} \tag{4.36}$$

Note that $\mathbf{z}$ is a standard Gaussian vector, $\theta_T^\top \mathbf{z}$ is distributed as $\mathcal{N}(0, \|\theta_T\|_2^2)$ Thus, Eq. (4.36) is equivalent to $\Phi(\frac{\theta_T^\top \boldsymbol{\mu}}{\sigma\|\theta_T\|_2})$.

By the inequality $1 - \Phi(x) \leq \exp\{-x^2/2\}$ for $x > 0$, then we have

$$\Phi(\frac{\theta_T^\top \boldsymbol{\mu}}{\sigma\|\theta_T\|_2}) \geq 1 - \exp\{-\frac{(\frac{\theta_T^\top \boldsymbol{\mu}}{\sigma\|\theta_T\|_2})^2}{2}\} \geq 1 - \exp\{-\frac{1}{200}(\frac{b_0}{(1+2\sigma)\sigma})^2\}$$

We denote $g(\sigma)$ by:

$$g(\sigma) = \frac{\mathrm{Erf}[\frac{1-r}{\sqrt{2}\sigma}]}{2(1+2\sigma)\sigma} + \frac{\exp\left(-\frac{(r-1)^2}{2\sigma^2}\right)}{\sqrt{2\pi}(1+2\sigma)},$$

where $g(\sigma) > 0$ for any $\sigma > 0$. Note that $g(\sigma) \to \infty$ when $\sigma \to 0$, and $g(\sigma)$ is monotone decreasing as σ increases since $g'(\sigma) < 0$ for $\sigma > 0$.

□

Lemma 4.D.1. *Let $X = (X_1, X_2, \ldots, X_n) \in \mathbb{R}^n$ be a random vector with independent, Gaussian coordinates X_i with $\mathbb{E}[X_i] = 0$ and $\mathbb{E}[X_i^2] = 1 < \infty$. Then*

$$\Pr[|\, \|X\|_2 - \sqrt{n}| \geq \sqrt{n}] \leq 2\exp(-an),$$

where $a > 0$ is a constant.

Proof. The Gaussian concentration result is taken from Proposition 5.34 in [130], which will be used here for proving Theorem 4.4.1. □

4.E Additional Learning Curves

We provide additional learning curves on DomainNet dataset, shown in Figure 4.5. The dataset contains 12 pairs of tasks showing: (1) target classifiers have higher prediction accuracy during the early-training time; (2) leverage ETP by using ELR can alleviate the memorization of unbounded noisy labels generated by source models.

4.F Experimental Details

Datasets. We use four benchmark datasets to verify the effectiveness of leveraging the early-time training phenomenon to address unbounded label noise. Office-31 [106] contains 4,652 images in three domains (Amazon, DSLR, and Webcam), and each domain consists of 31 classes. Office-Home [128] contains 15,550 images in four domains (Real, Clipart, Art, and Product), and each domain consists of 65 classes. VisDA [97] contains 152K synthetic images and 55K real object images with 12 classes. DomainNet [96] contains around 600K images in six different domains (Clipart, Infograph, Painting, Quickdraw, Real and Sketch). Following previous work [120, 74], we select 40 the most commonly-seen classes from four domains: Real, Clipart, Painting, and Sketch.

Implementation. We use ResNet-50 [43] for Office-31, Office-Home and DomainNet, and ResNet-101 [43] for VisDA as backbones. We adopt a fully connected (FC) layer as the feature extractor on the backbone and another FC layer as the classifier head. The batch normalization layer is put between the two FC layers and the weight normalization layer is implemented on the last FC layer. We set the learning rate to 1e-4 for all layers except for the last two FC layers, where we apply 1e-3 for the learning rate for all datasets. The training for source models are set to be consistent with the SHOT [71]. The hyperparameters for ELR with self-training, ELR with SHOT, ELR with G-SFDA, and ELR with NRC on four different datasets are shown in Table 4.5. We note that for ELR with self-training, there is only one

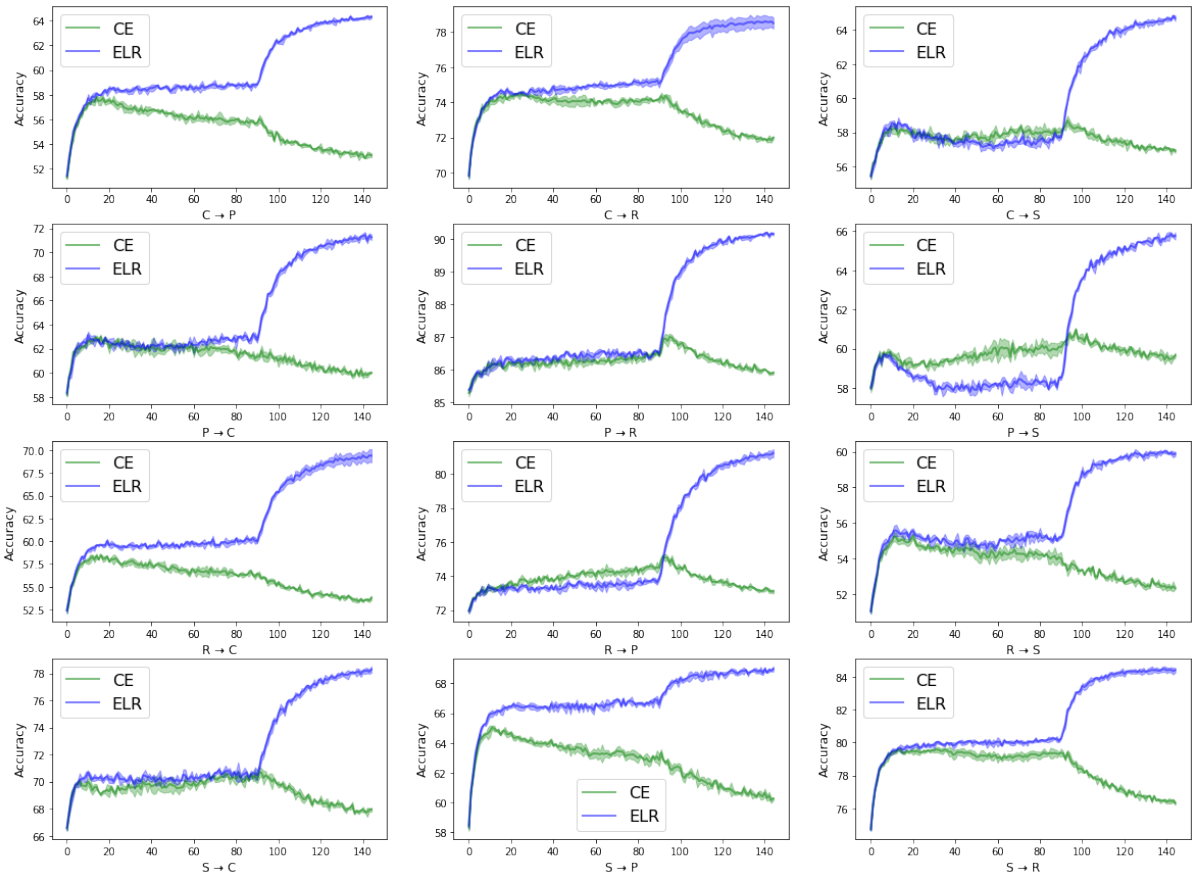


Figure 4.5: The source models are used to initialize the classifiers and annotate unlabeled target data. As the classifiers memorize the unbounded label noise very fast, we evaluate the prediction accuracy on target data every batch for the first 90 steps. After the 90 steps, we evaluate the prediction accuracy for every 0.3 epoch. We use the CE and ELR to train the classifiers on the labeled target data, shown in solid green lines and solid blue lines, respectively.

hyperparameter β to tune. The hyperparameters for existing SFDA algorithms are set to be consistent with their reported values for Office-31, Office-Home, and VisDA datasets. As these SFDA algorithms have not reported their performance for DomainNet dataset, We follow the hyperparameter search strategy from their work [71, 147, 148], and choose the optimal hyperparameters $\beta = 0.3$ for SHOT, $K = 5$ and $M = 5$ for NRC, and $k = 5$ for G-SFDA.

4.G Memorization Speed Between Label Noise in SFDA and in Conventional LLN settings

Although ETP exists in both SFDA and conventional LLN scenarios, the memorization speed for them is still different. Specifically, the target classifiers memorize noisy labels much faster in the SFDA scenario. It has already been shown that it takes many epochs before classifiers start memorizing noisy labels in conventional LLN scenario [75, 138]. We highlight that the

Table 4.5: Optimal Hypermaraters (β/λ) on various datasets.

Hyperparameters: β/λ	Office-31	Office-Home	VisDA	DomainNet
ELR only	0.9/–	0.99/–	0.99/–	0.9/–
ELR + SHOT	0.7/1.0	0.6/3.0	0.6/25	0.7/7.0
ELR + G-SFDA	0.8/1.0	0.9/1.0	0.5/7.0	0.8/12.0
ELR + NRC	0.5/1.0	0.6/3.0	0.5/3.0	0.8/3.0

main factor causing the difference is the label noise. To show it, we replace the unbounded label noise in SFDA with bounded random label noise, and we keep the other settings unchanged as introduced in 4.4. To replace the unbounded label noise with bounded random label noise, we use the source model to identify mislabeled target samples, then we assign random labels to these mislabeled samples. Figure 4.6 and Figure 4.7 show the learning curves on Office-Home and Office-31 datasets with unbounded label noise and random bounded label noise. To better visualize the learning curves with unbounded label noise, we re-plot Figures 4.6-4.7 with different y scale in Figures 4.8-4.9. These figures demonstrate that target classifiers memorizing noisy labels with unbounded label noise is much faster than noisy labels with random bounded label noise. The classifiers with bounded label noise (colored in red) are expected to memorize all noisy labels eventually. As illustrated in Figures 4.8-4.9, the classifiers with unbounded label noise (colored in green) show that the noisy labels are already memorized. We note that for the first 90 steps, the prediction accuracy is evaluated every batch, while the prediction accuracy is evaluated every 0.3 epoch after that time. Therefore, for unbounded label noise, target classifiers start memorizing the noisy labels within the first epoch (consisting of more than 90 batches).

There are some existing LLN methods such as PCL [160] to purify noisy labels every epoch based on ETP. Due to this difference, these LLN methods are not helpful to solving label noise in SFDA as they are not able to capture the benefits of ETP. Our empirical results in Section 4.5.1 can support this argument. We also note that PCL does not suffer from the fast memorization speed and is able to capture the benefits of ETP in conventional LLN settings. As we indicated in Figures 4.6-4.7, it takes much longer time (more than a few epochs) for target classifiers to start memorizing bounded noisy labels. We hope these insights can motivate the researcher to consider memorization speed and design algorithms better for SFDA.

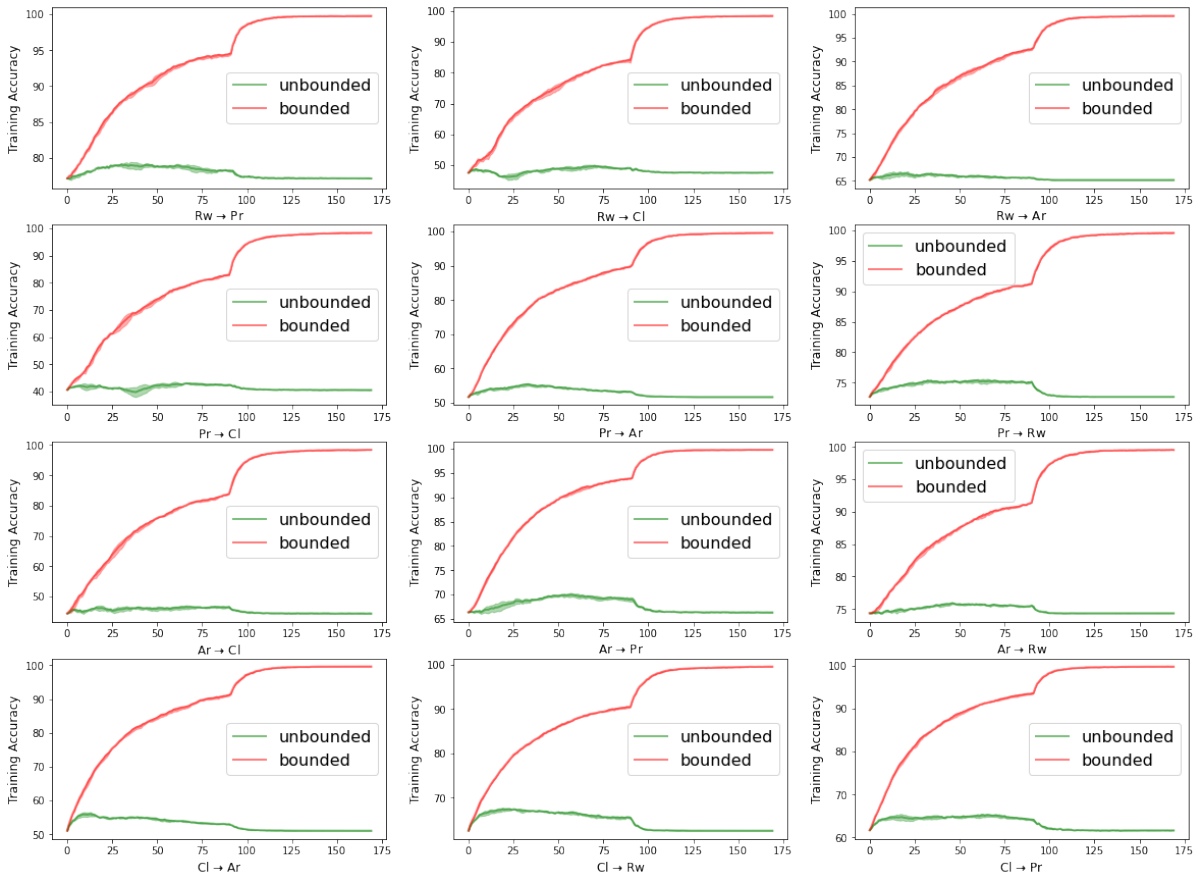


Figure 4.6: Training accuracy on Office-Home dataset. The solid green lines represent the unbounded label noise in SFDA, whereas the solid red lines represent the bounded label noise.

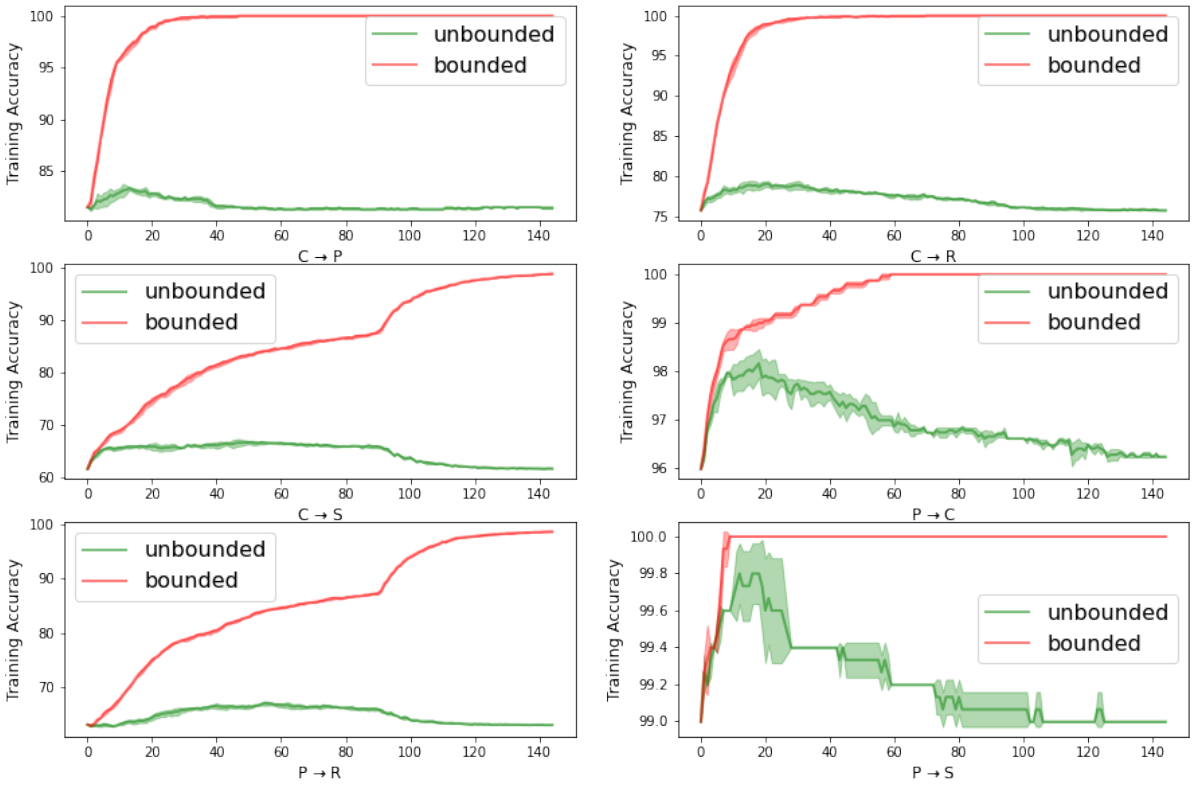


Figure 4.7: Training accuracy on Office-31 dataset. The solid green lines represent the unbounded label noise in SFDA, whereas the solid red lines represent the bounded label noise.

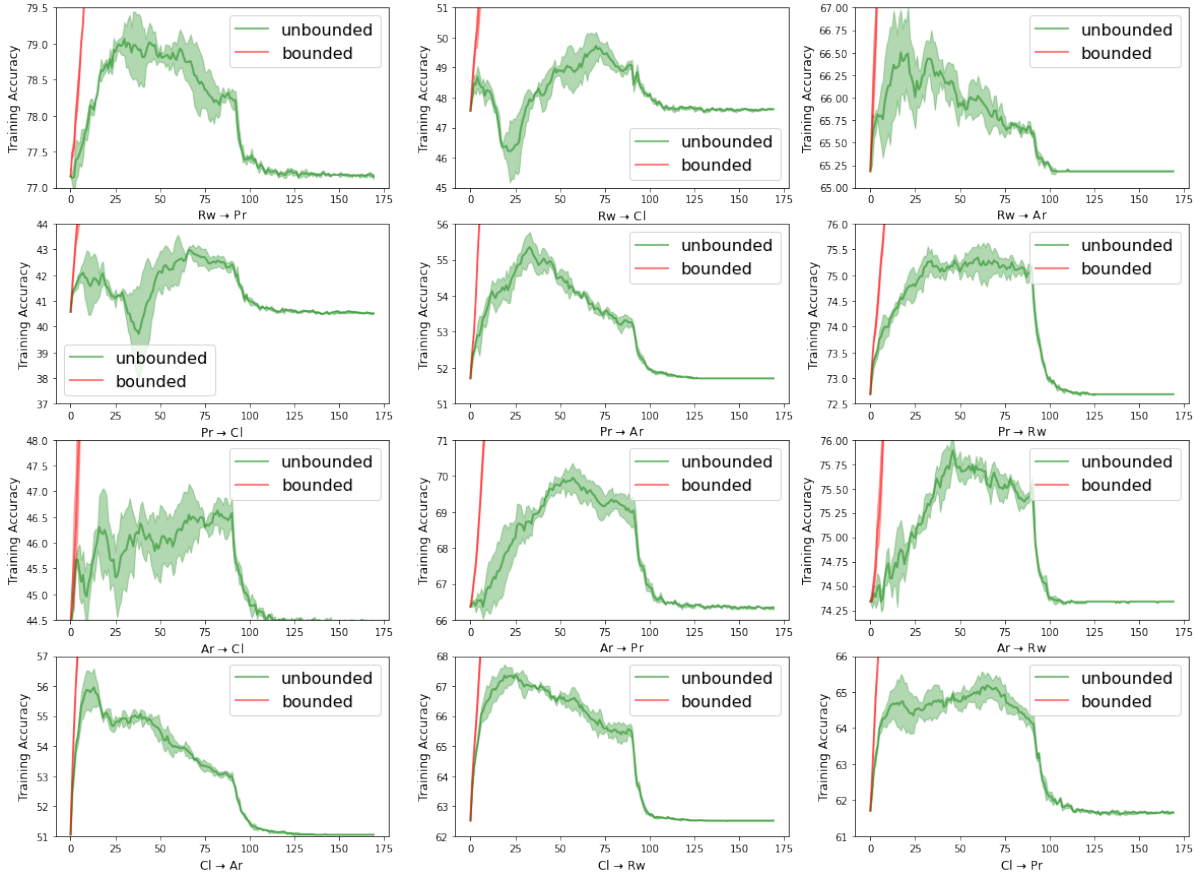


Figure 4.8: Figure 4.6 with different y-scale to better show learning details of the unbounded label noise.

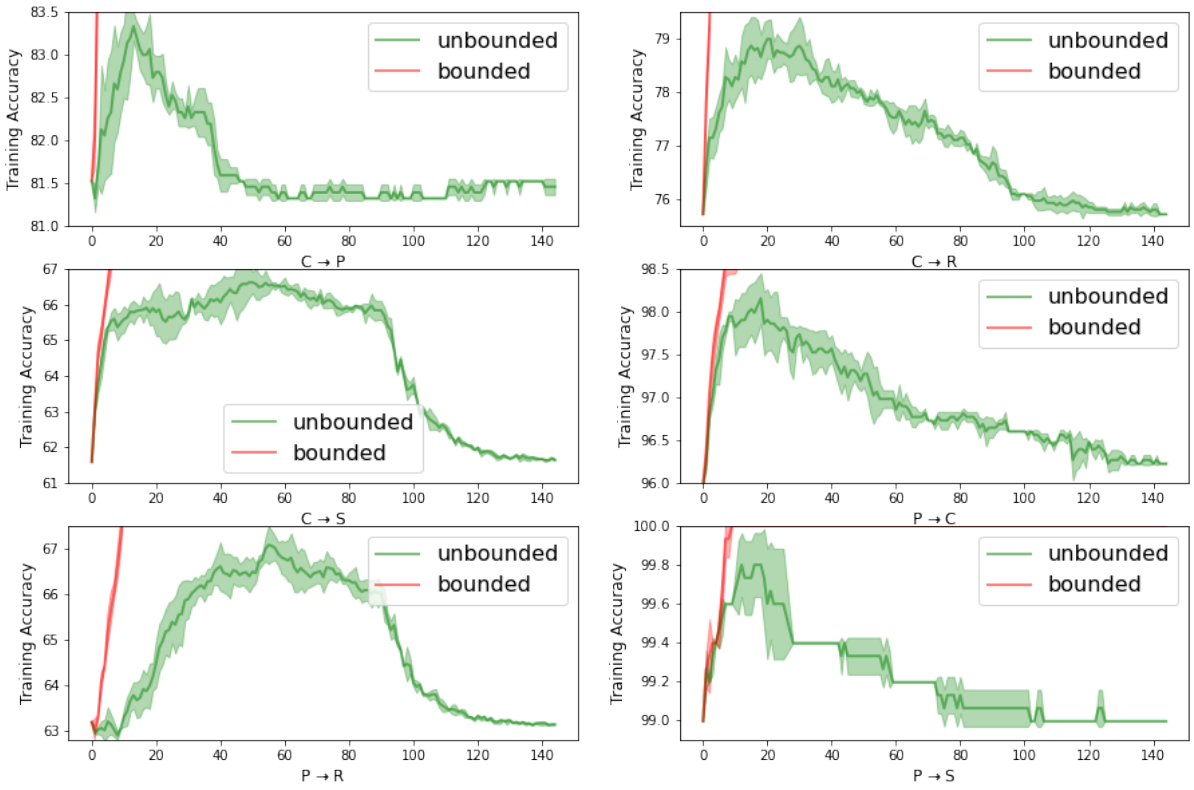


Figure 4.9: Figure 4.7 with different y-scale to better show learning details of the unbounded label noise.

Chapter 5

Conclusion and Future Work

Learning with noisy labels is a problem of great practical importance. In this thesis, we studied learning with noisy labels under various scenarios. First, we focused on learning noise-robust representations from examples with noisy labels. Second, we studied the benefits of self-supervised representations that are learned from examples without label information. Third, we connected the learning with label noise field to the source-free domain adaptation (SFDA) field and focused on studying the label noise in SFDA, whose distribution is quite different from conventional label noise settings. Below we will present the main contributions of each work.

In Chapter 2, we theoretically studied the robustness of learned contrastive representations, and we develop a novel algorithm to learn such representations from noisy training data without accessing true labels. In Chapter 3, we rigorously analyzed that the self-supervised representations, which are learned without accessing any label information, exhibit a better cluster structure that the linear classifier performs better on these representations. In Chapter 4, we solved the problem of SFDA and we showed that it can be formulated as the problem of learning with label noise. We also analyzed that the label noise in SFDA is quite different from label noise in conventional settings, and then we proposed a theoretically motivated approach to address it.

5.1 Future Work

5.1.1 Learning With Imperfect Data

With the recent emergence of large-scale datasets, noisy labels are not the only issue that make the performance of the model drop.

Imbalanced datasets are also the obstacles to obtaining reliable models [150, 46, 21]. A natural extension of learning with label noise is learning with both label noise and imbalanced data. The work related to this topic is very scarce. [9] studied both learning with label noise and learning with imbalanced data together. However, it only focuses on analyzing random label noise, which is not realistic in practice. The recent work about learning with label noise lies in studying more practically instance-dependent label noise [160, 18]. In future research, studying more realistic label noise in imbalanced datasets would be interesting and valuable.

Besides imbalanced datasets, out-of-distribution samples (OOD) also affects the model learning and they are very common, especially for datasets collected from the Internet. For example, the researcher aims to train a model to classify different fruits, and the training dataset is collected from the web. There are very likely that some images do not belong to any target categories. For example, a car with orange paint might be collected as an orange. When both OOD samples and mislabeled samples are in the training datasets, there should be two different procedures to deal with them independently. One is for correcting labels and another is for removing OOD samples. [67] is the one that considered both mislabeled samples and OOD samples by simply treating less confident samples as OOD samples. Since models also lack confidence about hard examples with clean labels, which are important for models to learn a good decision boundary, simply filtering out low-confident samples can degrade the models' performance. As a result, a better approach to address issues from both OOD samples and mislabeled can be explored in future research.

5.1.2 Beyond Deep Classification Tasks

This thesis focused on deep learning with classification tasks. In this setting, we considered that a label $y \in \mathcal{Y}$ can be incorrectly labeled by another label $\tilde{y} \in \mathcal{Y}$, where $\mathcal{Y}$ is a set of integers. Existing techniques for dealing with label noise focus on categorical labels. Given that many real-world tasks involve continuous target values, label noise also exists in deep regression tasks. There are many scenarios that are also common in the field of deep learning and are related to regression tasks.

Object detection and segmentation are another popular tasks in the field of deep learning, which require very intensive labeling work. Object detection should identify different objects and their locations in each image, and segmentation partitions an image into multiple pieces. For object detection, annotators should label the category of the object and its position, where both the category and the position might be mislabeled. That is $y \in \mathbb{Z} \times \mathbb{R}^4 \rightarrow \tilde{y} \in \mathbb{Z} \times \mathbb{R}^4$, where we assume $\mathbb{Z}$ is a set of integers. Since the position is a continuous variable, the problem can be formulated as learning with label noise for both classification and regression tasks. For segmentation, annotators should label each pixel in the image with a category label, where humans will more likely to mislabel pixels around the boundaries between different objects. That is $y \in \mathbb{R}^M \rightarrow \tilde{y} \in \mathbb{R}^M$, where M is the number of pixels (e.g. $M = 50176$ for 224 by 224 images). Given that the label space is changed drastically, it is a valuable direction to dig in.

5.1.3 Other Data Structures

This thesis mainly focused on image classification tasks. It would be very exciting to extend the image classification method to other data structures such as natural language, speech signals, and tabular data, where tabular data is organized in a table with rows being the id and columns being the features. As [4] indicated, deep neural networks are prone to fit images with simple patterns first from the noisy training data, where the simple patterns of images are highly correlated with images with clean labels. Most image classification methods implicitly or explicitly leveraged this observation. As this observation might not exist when we replace image data with tabular data, natural language or signal data, we cannot simply apply existing image classification methods to other data structures. The hybrid data structure is a more

complicated data structure, for example, which may consist of both images and language. To this end, it will be quite interesting to explore a unified approach that can address noisy label issues for both image data and other types of data.

Bibliography

[1] S. M. Ahmed, D. S. Raychaudhuri, S. Paul, S. Oymak, and A. K. Roy-Chowdhury. Unsupervised multi-source domain adaptation without access to source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10103–10112, 2021.

[2] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182. PMLR, 2016.

[3] E. Arazo, D. Ortego, P. Albert, N. O'Connor, and K. McGuinness. Unsupervised label noise modeling and loss correction. In *International Conference on Machine Learning*, pages 312–321. PMLR, 2019.

[4] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pages 233–242. PMLR, 2017.

[5] Y. Bai, E. Yang, B. Han, Y. Yang, J. Li, Y. Mao, G. Niu, and T. Liu. Understanding and improving early stopping for learning with noisy labels. *Advances in Neural Information Processing Systems*, 34, 2021.

[6] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

[7] S. Ben-David, J. Blitzer, K. Crammer, F. Pereira, et al. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19:137, 2007.

[8] A. Berthon, B. Han, G. Niu, T. Liu, and M. Sugiyama. Confidence scores make instance-dependent label-noise learning possible. In *International Conference on Machine Learning*, pages 825–836. PMLR, 2021.

[9] K. Cao, Y. Chen, J. Lu, N. Arechiga, A. Gaidon, and T. Ma. Heteroskedastic and imbalanced deep learning with adaptive regularization. *arXiv preprint arXiv:2006.15766*, 2020.

[10] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

[11] P. Chen, G. Chen, J. Ye, jingwei zhao, and P.-A. Heng. Noise against noise: stochastic label noise helps combat inherent label noise. In *International Conference on Learning Representations*, 2021.

[12] P. Chen, J. Ye, G. Chen, J. Zhao, and P.-A. Heng. Beyond class-conditional assumption: A primary attempt to combat instance-dependent label noise. *arXiv preprint arXiv:2012.05458*, 2020.

[13] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[14] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020.

[15] X. Chen, H. Fan, R. Girshick, and K. He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[16] X. Chen and K. He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[17] Y. Chen, X. Shen, S. X. Hu, and J. A. Suykens. Boosting co-teaching with compression regularization for label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2688–2692, 2021.

[18] H. Cheng, Z. Zhu, X. Li, Y. Gong, X. Sun, and Y. Liu. Learning with instance-dependent label noise: A sample sieve approach. *arXiv preprint arXiv:2010.02347*, 2020.

[19] M. Ciortan, R. Dupuis, and T. Peel. A framework using contrastive learning for classification with noisy labels. *Data*, 6(6):61, 2021.

[20] S. Cui, S. Wang, J. Zhuo, L. Li, Q. Huang, and Q. Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. *CVPR*, 2020.

[21] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[23] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[24] E. Englesson and H. Azizpour. Generalized jensen-shannon divergence loss for learning with noisy labels. *arXiv preprint arXiv:2105.04522*, 2021.

[25] F. Farnia and D. Tse. A minimax approach to supervised learning. *Advances in Neural Information Processing Systems*, 29:4240–4248, 2016.

[26] L. Feng, S. Shu, Z. Lin, F. Lv, L. Li, and B. An. Can cross entropy loss be robust to label noise? In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 2206–2212, 2021.

[27] K. Fukunaga. *Introduction to statistical pattern recognition*. Elsevier, 2013.

[28] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[29] A. Ghosh, H. Kumar, and P. Sastry. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[30] A. Ghosh and A. Lan. Contrastive learning improves model robustness under label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2703–2708, 2021.

[31] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[32] P. Goel and L. Chen. On the robustness of monte carlo dropout trained with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2219–2228, 2021.

[33] J. Goldberger and E. Ben-Reuven. Training deep neural-networks using a noise adaptation layer. 2016.

[34] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. Ieee, 2013.

[35] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5, 2009.

[36] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[37] X. Gu, J. Sun, and Z. Xu. Spherical space domain adaptation with robust pseudo-label loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9101–9110, 2020.

[38] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *arXiv preprint arXiv:1804.06872*, 2018.

[39] J. Z. HaoChen, C. Wei, A. Gaidon, and T. Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *arXiv preprint arXiv:2106.04156*, 2021.

[40] H. Harutyunyan, K. Reing, G. Ver Steeg, and A. Galstyan. Improving generalization by controlling label-noise information in neural network weights. In *International Conference on Machine Learning*, pages 4071–4081. PMLR, 2020.

[41] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[42] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[43] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[44] D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. *arXiv preprint arXiv:1802.05300*, 2018.

[45] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. PMLR, 2018.

[46] Y. Hong, S. Han, K. Choi, S. Seo, B. Kim, and B. 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.

[47] J. Howard and S. Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[48] W. Hu, Z. Li, and D. Yu. Simple and effective regularization methods for training on noisily labeled data with generalization guarantee. *arXiv preprint arXiv:1905.11368*, 2019.

[49] L. Huang, C. Zhang, and H. Zhang. Self-adaptive training: beyond empirical risk minimization. *Advances in neural information processing systems*, 33:19365–19376, 2020.

[50] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[51] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. 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.

[52] Y. Jin, X. Wang, M. Long, and J. Wang. Minimum class confusion for versatile domain adaptation. *ECCV*, 2020.

[53] I. Jindal, M. Nokleby, and X. Chen. Learning deep networks from noisy labels with dropout regularization. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 967–972. IEEE, 2016.

[54] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019.

[55] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020.

[56] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[57] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[58] A. Krogh and J. Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991.

[59] J. N. Kundu, N. Venkat, R. V. Babu, et al. Universal source-free domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4544–4553, 2020.

[60] J. N. Kundu, N. Venkat, A. Revanur, R. V. Babu, et al. Towards inheritable models for open-set domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12376–12385, 2020.

[61] J. Lafferty and L. Wasserman. Statistical analysis of semi-supervised regression. 2007.

[62] C.-Y. Lee, T. Batra, M. H. Baig, and D. Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10285–10295, 2019.

[63] K. Lee, S. Yun, K. Lee, H. Lee, B. Li, and J. Shin. Robust inference via generative classifiers for handling noisy labels. In *International Conference on Machine Learning*, pages 3763–3772. PMLR, 2019.

[64] K. Lee, Y. Zhu, K. Sohn, C.-L. Li, J. Shin, and H. Lee. I-mix: A domain-agnostic strategy for contrastive representation learning. *arXiv preprint arXiv:2010.08887*, 2020.

[65] K.-H. Lee, X. He, L. Zhang, and L. Yang. Cleannet: Transfer learning for scalable image classifier training with label noise. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5447–5456, 2018.

[66] J. Li, R. Socher, and S. C. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020.

[67] J. Li, C. Xiong, and S. C. Hoi. Mopro: Webly supervised learning with momentum prototypes. *arXiv preprint arXiv:2009.07995*, 2020.

[68] J. Li, C. Xiong, and S. C. Hoi. Learning from noisy data with robust representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9485–9494, 2021.

[69] M. Li, M. Soltanolkotabi, and S. Oymak. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *International conference on artificial intelligence and statistics*, pages 4313–4324. PMLR, 2020.

[70] W. Li, L. Wang, W. Li, E. Agustsson, and L. Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017.

[71] J. Liang, D. Hu, and J. Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039. PMLR, 2020.

[72] J. Liang, D. Hu, Y. Wang, R. He, and J. Feng. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[73] H. Liu, J. Z. HaoChen, A. Gaidon, and T. Ma. Self-supervised learning is more robust to dataset imbalance. *arXiv preprint arXiv:2110.05025*, 2021.

[74] H. Liu, J. Wang, and M. Long. Cycle self-training for domain adaptation. *Advances in Neural Information Processing Systems*, 34, 2021.

[75] S. Liu, J. Niles-Weed, N. Razavian, and C. Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *arXiv preprint arXiv:2007.00151*, 2020.

[76] T. Liu and D. Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2015.

[77] Y. Liu, W. Zhang, and J. Wang. Source-free domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1215–1224, 2021.

[78] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. *International Conference on Machine Learning*, 2015.

[79] M. Long, Z. Cao, J. Wang, and M. I. Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 1647–1657, 2018.

[80] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE international conference on computer vision*, pages 2200–2207, 2013.

[81] Z. Lu, Y. Yang, X. Zhu, C. Liu, Y.-Z. Song, and T. Xiang. Stochastic classifiers for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9111–9120, 2020.

[82] M. Lukasik, S. Bhojanapalli, A. Menon, and S. Kumar. Does label smoothing mitigate label noise? In *International Conference on Machine Learning*, pages 6448–6458. PMLR, 2020.

[83] Y. Lyu and I. W. Tsang. Curriculum loss: Robust learning and generalization against label corruption. *arXiv preprint arXiv:1905.10045*, 2019.

[84] S. Ma, Z. Zeng, D. McDuff, and Y. Song. Active contrastive learning of audio-visual video representations. *arXiv preprint arXiv:2009.09805*, 2020.

[85] X. Ma, H. Huang, Y. Wang, S. Romano, S. Erfani, and J. Bailey. Normalized loss functions for deep learning with noisy labels. In *International Conference on Machine Learning*, pages 6543–6553. PMLR, 2020.

[86] X. Ma, Y. Wang, M. E. Houle, S. Zhou, S. Erfani, S. Xia, S. Wijewickrema, and J. Bailey. Dimensionality-driven learning with noisy labels. In *International Conference on Machine Learning*, pages 3355–3364. PMLR, 2018.

[87] H. Maennel, I. Alabdulmohsin, I. Tolstikhin, R. J. Baldock, O. Bousquet, S. Gelly, and D. Keysers. What do neural networks learn when trained with random labels? *arXiv preprint arXiv:2006.10455*, 2020.

[88] E. Malach and S. Shalev-Shwartz. Decoupling" when to update" from" how to update". *arXiv preprint arXiv:1706.02613*, 2017.

[89] N. Manwani and P. Sastry. Noise tolerance under risk minimization. *IEEE transactions on cybernetics*, 43(3):1146–1151, 2013.

[90] B. Mirzasoleiman, K. Cao, and J. Leskovec. Coresets for robust training of neural networks against noisy labels. *arXiv preprint arXiv:2011.07451*, 2020.

[91] I. Misra and L. v. d. Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020.

[92] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari. Learning with noisy labels. *Advances in neural information processing systems*, 26:1196–1204, 2013.

[93] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[94] S. Ozair, C. Lynch, Y. Bengio, A. v. d. Oord, S. Levine, and P. Sermanet. Wasserstein dependency measure for representation learning. *arXiv preprint arXiv:1903.11780*, 2019.

[95] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. 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.

[96] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019.

[97] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and K. Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017.

[98] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017.

[99] S. Purushwalkam and A. Gupta. Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases. *arXiv preprint arXiv:2007.13916*, 2020.

[100] Z. Qiu, Y. Zhang, H. Lin, S. Niu, Y. Liu, Q. Du, and M. Tan. Source-free domain adaptation via avatar prototype generation and adaptation. *arXiv preprint arXiv:2106.15326*, 2021.

[101] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[102] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

[103] M. Ren, W. Zeng, B. Yang, and R. Urtasun. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, pages 4334–4343. PMLR, 2018.

[104] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[105] P. Rigollet. Generalization error bounds in semi-supervised classification under the cluster assumption. *Journal of Machine Learning Research*, 8(7), 2007.

[106] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.

[107] K. Saito, Y. Ushiku, T. Harada, and K. Saenko. Adversarial dropout regularization. *ICLR*, 2018.

[108] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2018.

[109] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[110] J. Shen, Y. Qu, W. Zhang, and Y. Yu. Wasserstein distance guided representation learning for domain adaptation. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[111] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[112] A. Singh, R. Nowak, and J. Zhu. Unlabeled data: Now it helps, now it doesn't. *Advances in neural information processing systems*, 21:1513–1520, 2008.

[113] H. Song, M. Kim, and J.-G. Lee. Selfie: Refurbishing unclean samples for robust deep learning. In *International Conference on Machine Learning*, pages 5907–5915. PMLR, 2019.

[114] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee. Learning from noisy labels with deep neural networks: A survey. *arXiv preprint arXiv:2007.08199*, 2020.

[115] D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, and N. Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

[116] K. Sridharan and S. M. Kakade. An information theoretic framework for multi-view learning. 2008.

[117] A. Srinivas, M. Laskin, and P. Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*, 2020.

[118] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[119] M. Sugiyama, M. Krauledat, and K.-R. Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(5), 2007.

[120] S. Tan, X. Peng, and K. Saenko. Class-imbalanced domain adaptation: an empirical odyssey. In *European Conference on Computer Vision*, pages 585–602. Springer, 2020.

[121] D. Tanaka, D. Ikami, T. Yamasaki, and K. 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.

[122] H. Tang, K. Chen, and K. Jia. Unsupervised domain adaptation via structurally regularized deep clustering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8725–8735, 2020.

[123] K. Tanwisuth, X. Fan, H. Zheng, S. Zhang, H. Zhang, B. Chen, and M. Zhou. A prototype-oriented framework for unsupervised domain adaptation. *Advances in Neural Information Processing Systems*, 34, 2021.

[124] Y. Tian, D. Krishnan, and P. Isola. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019.

[125] Y.-H. H. Tsai, M. Q. Ma, M. Yang, H. Zhao, L.-P. Morency, and R. Salakhutdinov. Self-supervised representation learning with relative predictive coding. *arXiv preprint arXiv:2103.11275*, 2021.

[126] Y.-H. H. Tsai, Y. Wu, R. Salakhutdinov, and L.-P. Morency. Self-supervised learning from a multi-view perspective. *arXiv preprint arXiv:2006.05576*, 2020.

[127] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[128] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017.

[129] V. Verma, T. Luong, K. Kawaguchi, H. Pham, and Q. Le. Towards domain-agnostic contrastive learning. In *International Conference on Machine Learning*, pages 10530–10541. PMLR, 2021.

[130] R. Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[131] R. Volpi, P. Morerio, S. Savarese, and V. Murino. Adversarial feature augmentation for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5495–5504, 2018.

[132] T. Wang and P. Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020.

[133] X. Wang, L. Li, W. Ye, M. Long, and J. Wang. Transferable attention for domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5345–5352, 2019.

[134] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. 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.

[135] H. Wei, L. Feng, X. Chen, and B. 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.

[136] J. Wei, H. Liu, T. Liu, G. Niu, M. Sugiyama, and Y. Liu. To smooth or not? when label smoothing meets noisy labels, 2021.

[137] Y. Wu, D. Inkpen, and A. El-Roby. Dual mixup regularized learning for adversarial domain adaptation. *ECCV*, 2020.

[138] X. Xia, T. Liu, B. Han, C. Gong, N. Wang, Z. Ge, and Y. Chang. Robust early-learning: Hindering the memorization of noisy labels. In *International Conference on Learning Representations*, 2020.

[139] X. Xia, T. Liu, B. Han, N. Wang, M. Gong, H. Liu, G. Niu, D. Tao, and M. Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. *Advances in Neural Information Processing Systems*, 33, 2020.

[140] X. Xia, T. Liu, N. Wang, B. Han, C. Gong, G. Niu, and M. Sugiyama. Are anchor points really indispensable in label-noise learning? *Advances in Neural Information Processing Systems*, 32:6838–6849, 2019.

[141] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. 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.

[142] E. Xie, J. Ding, W. Wang, X. Zhan, H. Xu, P. Sun, Z. Li, and P. Luo. Detco: Unsupervised contrastive learning for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8392–8401, 2021.

[143] R. Xu, G. Li, J. Yang, and L. Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[144] R. Xu, P. Liu, L. Wang, C. Chen, and J. Wang. Reliable weighted optimal transport for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4394–4403, 2020.

[145] Y. Xu, P. Cao, Y. Kong, and Y. Wang. L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. In *NeurIPS*, pages 6222–6233, 2019.

[146] G. Yang, H. Xia, M. Ding, and Z. Ding. Bi-directional generation for unsupervised domain adaptation. In *AAAI*, pages 6615–6622, 2020.

[147] S. Yang, J. van de Weijer, L. Herranz, S. Jui, et al. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. *Advances in Neural Information Processing Systems*, 34, 2021.

[148] S. Yang, Y. Wang, J. van de Weijer, L. Herranz, and S. Jui. Generalized source-free domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8978–8987, 2021.

[149] Y. Yang and Z. Xu. Rethinking the value of labels for improving class-imbalanced learning. *arXiv preprint arXiv:2006.07529*, 2020.

[150] Y. Yang, K. Zha, Y. Chen, H. Wang, and D. Katabi. Delving into deep imbalanced regression. In *International Conference on Machine Learning*, pages 11842–11851. PMLR, 2021.

[151] Y. Yao, T. Liu, B. Han, M. Gong, J. Deng, G. Niu, and M. Sugiyama. Dual t: Reducing estimation error for transition matrix in label-noise learning. *arXiv preprint arXiv:2006.07805*, 2020.

[152] K. Yi and J. Wu. Probabilistic end-to-end noise correction for learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7017–7025, 2019.

[153] L. Yi, S. Liu, Q. She, A. I. McLeod, and B. Wang. On learning contrastive representations for learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16682–16691, 2022.

[154] X. Yu, B. Han, J. Yao, G. Niu, I. Tsang, and M. Sugiyama. How does disagreement help generalization against label corruption? In *International Conference on Machine Learning*, pages 7164–7173. PMLR, 2019.

[155] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021.

[156] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations, ICLR 2017*, 2017.

[157] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[158] Y. Zhang, T. Liu, M. Long, and M. Jordan. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*, pages 7404–7413, 2019.

[159] Y. Zhang, H. Tang, K. Jia, and M. Tan. Domain-symmetric networks for adversarial domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5031–5040, 2019.

[160] Y. Zhang, S. Zheng, P. Wu, M. Goswami, and C. Chen. Learning with feature-dependent label noise: A progressive approach. *arXiv preprint arXiv:2103.07756*, 2021.

[161] Z. Zhang and M. R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *32nd Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

[162] Z. Zhang, H. Zhang, S. O. Arik, H. Lee, and T. Pfister. Distilling effective supervision from severe label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9294–9303, 2020.

[163] X. Zhao, R. Vemulapalli, P. A. Mansfield, B. Gong, B. Green, L. Shapira, and Y. Wu. Contrastive learning for label efficient semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10623–10633, 2021.

[164] Z. Zhu, T. Liu, and Y. Liu. A second-order approach to learning with instance-dependent label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10113–10123, 2021.

Curriculum Vitae

Name:	Li Yi
Post-Secondary Education and Degrees:	PhD in Statistics, 2018-2022 The University of Western Ontario, London, ON, Canada 2017 - 2018 M.Sc. in Statistics The University of Western Ontario, London, ON, Canada 2013-2017 B.Sc. in Statistics Southwestern University of Finance and Economics, Chengdu, SC, China
Related Work Experience:	Teaching Assistant The University of Western Ontario 2018 - 2021

Publications:

Li Yi, Sheng Liu, Qi She, A Ian McLeod, and Boyu Wang. On learning contrastive representations for learning with noisy labels. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16682–16691, 2022.