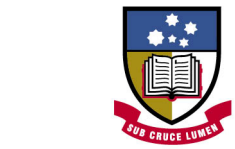


THE UNIVERSITY
of ADELAIDE

The Utility of Validation Sets for Meta-learning Methods for Noisy-Label and Imbalanced Learning Problems

by

Hoang Anh Dung

In fulfilment of the requirements for the degree of

Master of Philosophy

March 2023

Faculty of Sciences, Engineering and Technology

School of Computer and Mathematical Sciences

The University of Adelaide

Contents

List of Tables

List of Figures

Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

Hoang Anh Dung 21/3/2023

Acknowledgements

Master of Philosophy is an important, rigorous, challenging, and at the same time, an exciting milestone of my university life. Such experience not only has played an indispensable role in developing my career, intellectual and academic skills, but at the same time has dramatically modeled and expanded my perspective about academia, inspiring me to achieve many future goals. The completion of my thesis would not be possible without the help of my supervisors, friends and family members whom I am fortunate enough to be able to learn, work and collaborate with. Since it is impossible to give my thanks to all of them individually in the thesis, I would love to give my acknowledge to those friends/collaborators that contribute the most to my study.

First and foremost I am extremely grateful to my supervisors, Prof. Gustavo Carneiro and Dr. Vasileios Belagiannis for their valuable advices, continuous support, and patience during my MPhil. Their immense knowledge and plentiful experience have encouraged me all the time to pursue my academic research, from a student lacking research experience to a researcher with academic skills in computer vision.

I would also like to thank Dr. Cuong Nguyen for his writing support for my paper. Cuong has also indirectly provided great advice and suggestions about the research with insightful discussions and encouragement, in which his expertise and experience have helped me overcome many obstacles in my research. I am very grateful to have

him as my friend and colleague.

I would like to thank many colleagues at the Australian Institute for Machine Learning (AIML) for many amusing moments. You all have made my days at AIML more enjoyable.

I would like to thank many friends outside of work, especially Timothy and Cade, for their emotional support that helped me relieve stress during the time I spent in Australia. I have really enjoyed the time we spent on discussions about not only scientific research, but also political issues, future technologies, hobbies and many other things.

I would like to thank all the technical members at AIML. It is their kind help and tech support support that have made my study and life in the University of Adelaide a wonderful time.

Finally, I would like to thank all my friends, my parents and my younger sister. Without their tremendous understanding and encouragement in the past few years, it would have been impossible for me to complete my MPhil.

Abstract

In recent years, the world has witnessed successful developments for solving visual learning tasks, including image classification, object detection, and semantic segmentation. In large part, this success is due to the introduction of sophisticated deep learning models. Unfortunately, these models often require a massive amount of annotated training data in order to achieve acceptable performance. Annotating such large amount of data is not only time-consuming and costly, but also impractical or even impossible in many scenarios. Such problems have motivated the development of more affordable solutions, e.g., using crow-sourcing for data annotation process, which is a less expensive way to collect and annotate data, but may result in training dataset that are likely to be contaminated with label noise. Unfortunately, deep neural networks with their high capacity can easily overfit to those training samples, resulting in a deterioration in terms of prediction performance. Moreover, the presence of noisy-labeled data can aggravate label distribution imbalances on such training sets. Consequently, the field has intensively worked in the development of methods to address the issues produced by imbalanced noisy-label datasets in the training of deep learning models.

Many approaches have been proposed to handle training datasets with label noise and imbalanced label distribution. Among those, meta-learning has been demonstrating to be one of the most successful methods. Conventionally, a clean and balanced

validation set is usually required to train traditional meta-learning model. However, obtaining such validation set can be expensive or even impossible to access for certain datasets, particularly when the number of classes in the dataset is in the order of 10^3 or more. Such issues when building a clean dataset have motivated the development of meta-learning methods that automatically select validation samples that are likely to have clean labels and balanced class distribution. The aim is to form an "informative" validation set where the samples belonging to that set not only are clean and class-balanced, but also have high utility for the meta-learning algorithm. This is, however, missing from the majority of existing studies in meta-learning literature. In addition, a common problem with these methods is that when the level of label noise is high, most prior meta-learning methods are prone to overfitting due to their inability to select truly clean samples for the validation set.

The main focus of this thesis is, therefore, the proposal of a new meta-learning method that is robust to training sets that contain imbalanced class distribution and noisy labels, without requiring a clean and balanced validation set. The main technical contribution is the "informativeness" measure derived from a theoretical observation in the meta-learning approach, called Learning to Re-weight (L2W) which allows us to define a sample informativeness measure. Using this theoretical observation, the proposed method can automatically a highly-informative validation set that has highly-informative samples which have clean labels with high probability, where the class distribution is balanced. Empirical evaluation is then carried out on publicly available noisy label benchmarks that explore all common types of label noise, such as symmetric, asymmetric, instant-dependent, close-set, and open-set, on both synthetic and real-world datasets. The proposed method shows state-of-the-art performance on the majority of these benchmarks and outperforms all previous meta-learning approaches

by a large margin.

In summary, the newly-proposed meta-learning method has replaced the manual data collection and annotation to form a validation set by an automatic mechanism, while substantially boosting their prediction performance on several benchmarks. Despite its affordability and effectiveness, the proposed method still has some drawbacks, especially the overfitting issue at extremely-high label noise. Such weakness will be investigated and studied as a part of my future work.

Chapter 1

Introduction

This chapter briefly introduces the problems of noisy-label and imbalanced learning, presenting relevant background information and existing open research problems in these fields. It is then followed by a few prominent approaches that contain important research directions for noisy-label and imbalanced learning which motivated the development of our method. In particular, the strengths and limitations of each approach are discussed in details to identify the knowledge gap and potential improvements that I aim to address. Additionally, the aims and objectives of our research are presented, together with our contribution. We conclude the chapter with an outline of the structure of this thesis.

1.1 Motivation to noisy-label learning problems

Within the past decade, there have been great advancements in computer vision partly because of the development of deep learning models. Such successes often rely on extremely large learning capacity models and their potential to generalize well when

trained with enormous amounts of high-quality training data. Those models have been widely adopted to build state-of-the-art image classifiers [52, 70, 157], object detectors, and semantic segmentation systems. However, the condition that training requires high-quality training data plays a crucial role in the performance of deep learning models, which drove substantial efforts to the collection and labeling of large-scale training sets, such as ImageNet [33] and COCO [96]. Unfortunately, to obtain such a large amount of annotated data is, however, arduous, costly and even infeasible in some situations. For instance, in some fields such as medical image analysis [2] and satellite image analysis [101], it is challenging to obtain large-scale and high-quality datasets due to privacy concerns and the cost associated with the collection and labeling processes. Additionally, data collection and labeling of medical image datasets [48, 124] require a large number of professionals in the medical field with extraordinary expertise to visually identify different diseases. However, these experts are often well-paid and time poor, which complicates the dataset collection and labeling. In addition, concerns with patient information privacy can also hinder the dataset construction. Another typical example is the collection and annotation of satellite images [195, 199], where a single high-resolution image can cost thousands of dollars, and some of these images may not be allowed to become publicly available due to the presence of sensitive information. As a result, these issues have inspired the computer vision and machine learning communities to develop data collection and labeling methods that are more affordable and logistically less complicated.

Crowd sourcing [14, 61, 141] is a method designed to alleviate the issue of relying on an expensive labeling process. However, such solution has a problematic downside, which is the fact that the labels produced by the cheap and mediocre crowd sourcing annotators may be incorrect. Consequently, it is likely that the data labels from crowd

sourcing have not only label noise, but also labeling biases and inconsistencies. Unfortunately, training deep learning models with such poorly curated datasets can lead to overfitting and poor generalization. In order to formulate the noisy label problems and compensated for the lack of available realistic noisy datasets, as well as simulating their impact over deep neural network training, many popular datasets such as CIFAR10, CIFAR100 [86] contaminated with synthetic and artificial noise have been introduced as the standard benchmarks to study this problem. Additionally, a wide range of different types of noise have also been introduced, such as symmetric noise, asymmetric noise, open-set noise, semantic noise and real world noise, which we will provide definition in detail in the next Chapter.

Fig. 1.1 illustrates the consequences of training a binary deep neural network classifier to fit training sets contaminated with two different types of label noise, namely uniform noise (a.k.a. symmetric noise, or random noise), and semantic noise (noise that is generated by an imperfect annotator, e.g., a classifier trained on noisy data). We can see that the presence of label noise (on the right) will bias the clean-label dataset classification boundary (on the left), so it can fit the noisy-label datasets. Such biased classification boundary will lead to low classification accuracy on the test set. Moreover, existing noisy label learning frameworks often demand hyper-parameter tuning or extra clean data as anchor points to handle confirmation bias, which either increase the data cost further or make those frameworks less robust to unseen noisy datasets.

1.2 Class-Imbalanced learning problem

Imbalanced learning [63, 72, 148, 152, 156] is a challenging issue that has captured the attention of the computer vision and machine learning communities due to its preva-

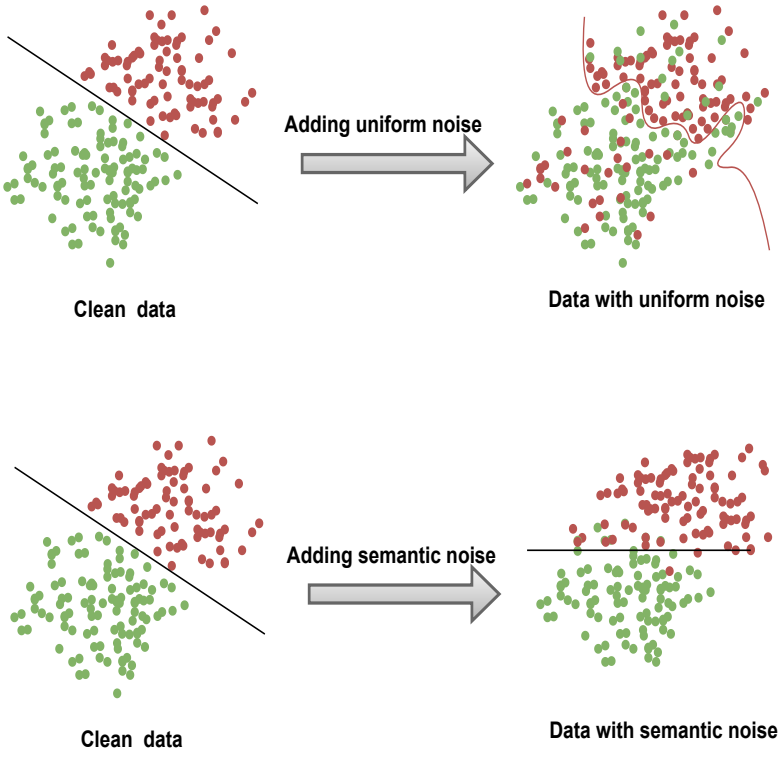


Figure 1.1: Impact of a noisy-label dataset for training a binary classifier, with the black line representing a classification boundary learned with the corresponding data (class 1:red points, class 2: green points). The upper part illustrates the classifier training when the dataset is contaminated with uniform (or symmetric) label noise, while the lower part shows the negative influence of semantic noise, obtained by flipping the label of samples that lie near the class boundary.

lence in real-world datasets. The problem is defined by large differences in the number of samples per class available for training, which form majority classes (i.e., classes that contain a relatively large amount of training samples) and minority classes (i.e., classes that have a relatively small amount of training samples) [193]. Such imbalanced learning problem can severely degrade classification accuracy of deep learning models, particularly for the minority classes. A typical cause of imbalanced learning lies in the natural prevalence of classes in certain classification problems. For example,

Figure 1.2: Unfavorable impact of imbalanced data on the training of a binary classification model (blue points: class 1, red points: class 2), with the black line representing the classification boundary learned with the corresponding datasets. We can see that the classification boundary trained with an imbalanced dataset (on the right) will favor the majority class, compared to the classifier trained with a balanced training (on the left).

datasets available for medical image analysis tasks [17, 198] are usually plagued with imbalanced distributions because disease class samples tend to be (thankfully) much rarer than the normal class samples. In addition, certain diseases can be rarer than other diseases [77], leading to minority classes with extremely few samples. Another potential cause of data distribution imbalance is bias in the data collection and labeling processes [11, 118]. For instance, under a domain-specific data collection process, samples can be acquired in a biased manner following a skewed prior knowledge. Also, data can be annotated with wrong labels because of annotator's prejudices. These factors are commonly observed in real-world datasets, exacerbating the imbalanced learning problem, which can have an unfavorable impact on the training of deep learning models. Fig. 1.2 illustrates the negative influence of training with imbalanced input data, which causes the classifier to reduce the risk of classifying incorrectly samples from the majority classes because of their larger contribution to the classification loss.

The consequence of traditional classification loss minimization using such imbalanced training sets is the low classification accuracy for the minority classes, particularly in comparison with the classification performance of the majority classes. Therefore, many solutions have been devised to alleviate the negative impact of this data bias problem [63, 72, 148, 152, 156]. However, since most of previous imbalanced learning papers assume that the training set contains only clean labels, there are not many previously published papers that attempt to address imbalanced learning and noisy-label learning at the same time. Moreover, as most of the methods designed for noisy-label learning and imbalanced learning have strong assumptions, such as a balanced training set or a clean-label training set, these methods can suffer from performance degradation when dealing with the combined problems of noisy-label and imbalanced learning. These issues will be discussed in more details in the next Chapter.

1.3 Background and motivation

A great number of methods have been designed for noisy-label learning problems, which can be roughly grouped as follows: ensemble learning [113], student-teacher model [147], robust loss functions [102, 155, 159], , label cleaning [68, 183], iterative label correction [194], co-teaching [52, 70, 92, 110, 179], dimensionality reduction [108], semi-supervised learning [92, 121, 122], meta learning [1, 2, 53, 70, 131, 139, 144, 149, 171, 196, 197], and mixed methods [71, 81, 120, 181, 190]. However, the experimental setting used by the methods above usually relies on a training set containing a balanced distribution. Consequently, most of these methods suffer significant performance degradation in imbalanced learning problems, except for the meta learning approaches [1, 2, 70, 131, 139, 149, 171, 196, 197]. This is because the meta objective

of the meta learning framework for noisy label problems relies on a balanced clean validation set to guide the training of the model. Since this clean validation set is always balanced, samples from minority classes in imbalanced learning setting will receive a larger weight under the meta learning framework, which will alleviate the issues caused by the imbalanced learning problem.

On the other hand, the large attention provided by the scientific community to the imbalanced learning problem has also resulted in many effective techniques [24, 100, 153, 160, 164], which can be divided into several groups, such as: [193]: transfer learning [24, 160], classifier design [100, 164], re-sampling (e.g., meta learning) [153], decoupled training [72, 74], ensemble learning [45, 203], cost-sensitive learning [36, 145, 204], data augmentation [23, 185], logit adjustment [112, 127] and representation learning [66, 191]. Regrettably, most of the existing imbalanced learning methods assume the input training data to only have clean labels, reducing their practicability and performance when dealing with the datasets contaminated with label noise. The only technique that does not make such assumption is meta learning, which, as already described above, can also handle noisy-label learning by using a balanced and clean-label validation set. Other imbalanced learning approaches have also been adopted for noisy-label learning (e.g., logit adjustment [112, 127] and classifier design [26, 92, 100, 122, 164]), but they often require a great deal of hyper-parameter tuning [26, 92] to become robust against specific types of label noise.

Meta learning is a versatile learning paradigm that can address many problems, such as imbalanced noisy-label learning and few-shot learning [64] problems. This type of learning optimizes the parameters of a model to be applied in some specific task (e.g., classification) and also the meta-parameters to improve model training. For instance, in noisy-label meta learning [1, 2, 70, 131, 139, 149, 171, 196, 197], the model

parameters are the neural network weights, while the meta-parameters usually consist of training sample weights and pseudo labeling weight [196, 197]. The meta learning methods optimize the model based on a weighted cross entropy loss that automatically down-weightthe losses of noisy samples and up-weight the losses of clean samples. This meta-parameter for training samples is iteratively optimized per batch with respect to the model's evaluation on an small clean-label dataset. For example, L2LWS [32] and CWS [31] approaches rely on a target deep neural network (DNN) and a meta-DNN that is pre-trained on a small clean validation dataset to re-weight the training samples to optimize the target DNN. Furthermore, learning to re-weight (L2W) [131] weights training samples based on the performance of one-step-ahead model on the validation set.

Most of previous meta learning methods require a manually labeled clean validation set, which is either expensive to acquire or unavailable in some real-world scenarios. Only recently, there have been several meta learning works such as FSR [196] and FAMUS [171] that attempt to combat this problem by exploiting an automatic mechanism to select the validation data. Hence, similarly to [171, 196], I aim to remove the necessity of a manually collected validation set. However, compared to previous studies [171, 196], the proposed method is motivated by the meta re-weighting optimization in order to select samples that are clean, informative and balanced. As FSR and FAMUS pay more attention to model simplification and cost reduction, they do not fully exploit the validation data for meta learning. As a result, their performance is not competitive compared to previous state-of-the-art meta learning frameworks. This remains a gap knowledge that needs to be resolved. In contrast, the proposed method, to the best of my knowledge, is the first work that leverages the noisy training dataset to build an informative validation set. Additionally, my work is one of the noisy label learning

pioneers that attempts to tackle the combination problem including both noisy-label and imbalanced learning.

1.4 My Contributions

As motivated by the shortcomings of previous meta learning methods designed to solve imbalanced learning and noisy-label learning, in this thesis,I propose several extensions of such meta learning methods to improve their classification performance. My contributions are summarized as follows:

- A set of new sample selection criteria to select informative, clean-labeled, and balanced samples to be included in the "pseudo-clean" validation set of a meta learning optimization;

- A novel meta learning algorithm that is designed to automatically constructs a "pseudo-clean" validation set by maximizing its utility according to the newly-proposed criteria, comprising the following steps: 1) detecting and labeling samples classified as clean-labeled from the noisy training set; 2) building of the validation set using the proposed utility criteria; and 3) meta learning using the validation set from step (2);

- Empirical demonstration the effectiveness of the proposed meta learning method, e.g., that our meta learning algorithm shows better classification accuracy than state-of-the-art meta learning and noisy-label learning approaches on most of the current public benchmarks in noisy-label learning and imbalanced learning.

1.5 Thesis Overview

This thesis is organized as follows:

- Chapter 1 briefly introduces the noisy-label learning and imbalanced learning problems, the knowledge gap that we aim to address, our research motivation, our contributions, and the general thesis outline.
- In Chapter 2, I present the literature review of related methods and defines the noisy-label learning and imbalanced learning problems in more detail. The strengths and weaknesses of each method are also analyzed with regards to the knowledge gap that I aim to handle.
- Chapter 3 introduces the main contributions of this thesis, which is a new meta learning algorithm and a new set of criteria to build a pseudo-clean validation set. The proposed algorithm is then evaluated on several benchmarks containing different types of label noise and class-imbalanced ratio. The effect of each components in the proposed algorithm is also analyzed and discussed via ablation studies.
- Chapter 4 summarizes all technical contributions of this thesis and discusses the weaknesses and potential improvements for the proposed method, as well as plans for future works.

Chapter 2

Literature review and related methods

In this chapter, I introduce the problems of noisy label learning and imbalanced data learning in detail. Initially, I define the noisy-label learning problem and explain different types of label noise and their potential negative impact on the training process. Subsequently, I review several noisy-label learning approaches, including their advantages and disadvantages and highlighting current research gaps. Following that, an introduction and a literature review of the imbalanced learning problem are presented. Then, we review other methodologies that were not explicitly designed for imbalanced learning or noisy-label learning, but are related to both problems or helpful for the development of this thesis. For instance, I discuss a few recent active-learning approaches that are relevant for the selection of validation samples for meta learning models, and self-supervised/feature representation learning that can potentially be used for imbalanced and noisy-label learning. I conclude the chapter with a discussion about previous works on meta learning classification based on sample re-weighting.

2.1 Noisy-label learning problems

As explained in Section 1.1, noisy-label datasets are now widespread in computer vision and machine learning. These datasets have driven these fields to propose new methods that can fully explore them. These methods are based on various techniques, which can be roughly classified as follows: robust loss functions [154, 159], label cleaning [68, 183], meta learning [53, 131], ensemble learning [113], and other methods [81, 190]. In conjunction with the development of methodologies to tackle poorly annotated datasets, many types of noise have been proposed and studied, leading to the implementation of noisy-label learning benchmarks [10, 25, 82, 125, 157, 180, 188]. In general, the noisy-label learning problems can be split into closed-set noise and open-set noise, which can then be sub-divided as symmetric (or uniform) noise, asymmetric noise, semantic noise, and instance-dependent noise, which are explained in detail below.

2.1.1 Closed-set versus open-set label noise

The noisy-label learning problem can also be divided into two categories: closed-set and open-set. A dataset is contaminated with closed-set label noise when all images in the dataset belong to one of the training classes. For example, on one hand, a problem is considered to be closed-set when all images in the dataset belong to one of the C classes that form the training set, but the labeling process may have introduced mistakes, where some of the labels have been switched to another label within the set of C classes. On the other hand, in open-set noise scenarios, the dataset images may belong to classes that are not in the set of C training classes, but their labels are one of the C classes. For instance, Clothing1M dataset [168] have images that are annotated with 14 training labels, despite containing images from visual classes outside

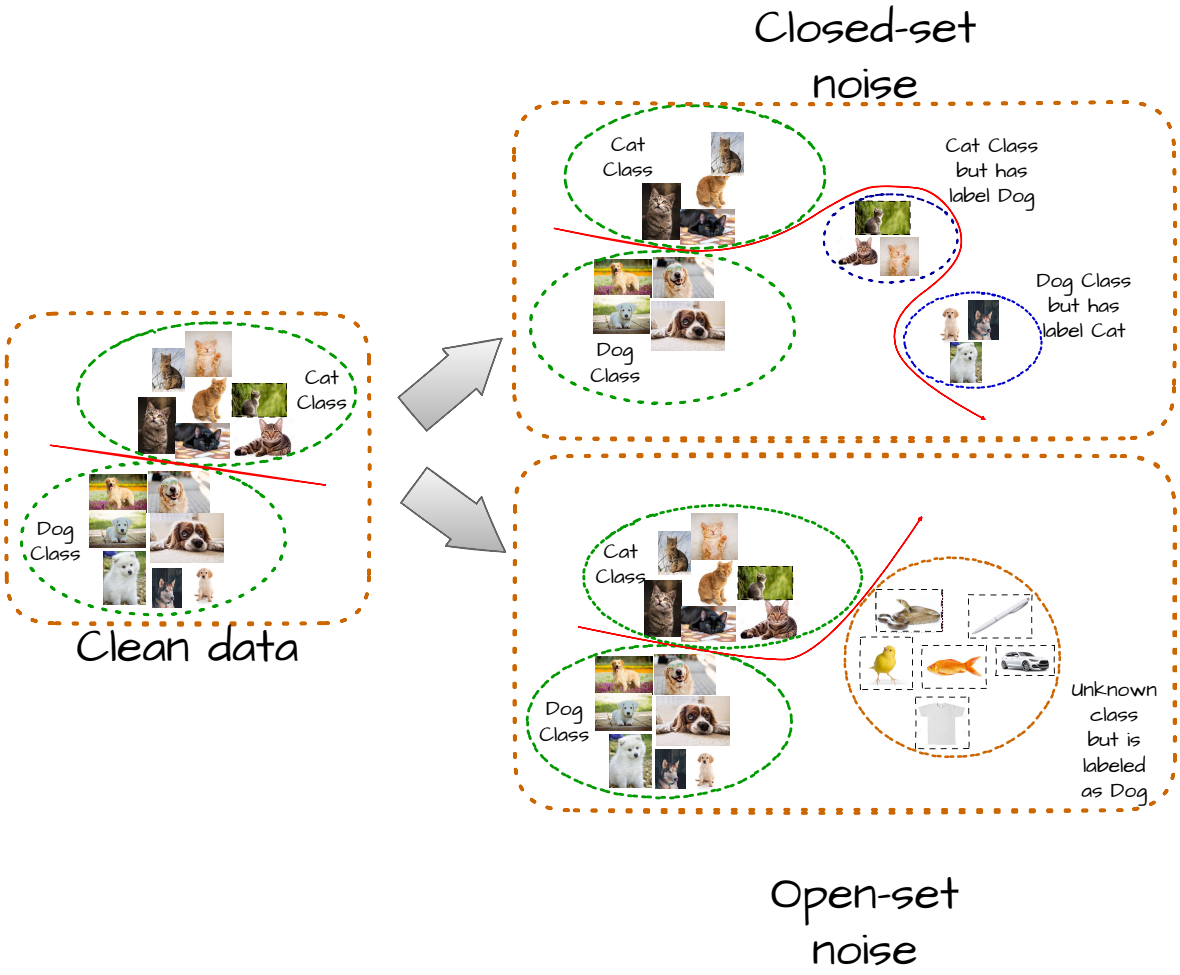


Figure 2.1: Illustration of the differences between closed-set and open-set noise. The red line denotes the trained classification boundary using the corresponding training data.

these 14 training classes. Open-set noise is common in scenarios where the data are indiscriminately collected from open sources, such as automatic web search engines (e.g., Google Images). Fig. 2.1 illustrates the differences between these two label noise problems.

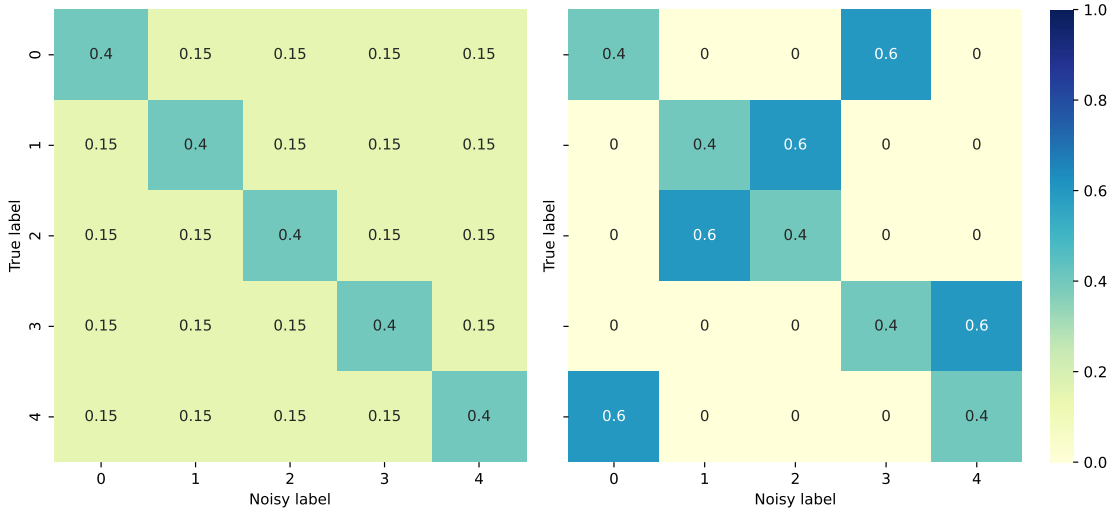


Figure 2.2: Symmetric (left) and asymmetric (right) noisy-label transition matrices in a classification problem containing five classes. The matrix on the left represents a symmetric label noise with rate 60%, uniformly distributed among the incorrect classes. The matrix on the right denotes an asymmetric label noise again with rate 60%, but with specific probabilities among training classes.

2.1.2 Symmetric noise

Symmetric noise is arguably the simplest type of label noise, where the probability that a label flips to any of the wrong classes follows a uniform distribution among the wrong training classes. In practice, symmetric noise is synthetically generated using a transition probability matrix, like the one shown in Fig. 2.2 (left), where each column shows a categorical distribution used to label samples of each class. Such symmetric noise happens due to encoding or transmission mistakes, and generally not because of labeling mistakes. Despite being unrealistic, the symmetric noise is an important benchmark for noisy-label learning evaluation.

2.1.3 Asymmetric noise

Being firstly introduced by Blum and Mitchell [13], the asymmetric noise is a more realistic class dependent type of noise that flips labels between similar classes. An example of asymmetric label noise is the CIFAR10 asymmetric noisy-label benchmark that flips semantically related labels, such as TRUCK and AUTOMOBILE, BIRD and PLANE, or DEER and HORSE. In practice, asymmetric noise can be simulated with a synthetic label generation using a transition probability matrix similar to the one in Fig. 2.2 (right), where each column shows a categorical distribution used to label samples of each class.

Similar to the symmetric noise, the asymmetric noise is still not very realistic because it purely depends on the classes, meaning that all images of a particular class will use the same categorical distribution to synthesize their noisy labels. Note that in real-world scenarios, we would expect certain images to be more likely to have their labels flipped given the ambiguity displayed by the images. An example of potential label uncertainty is depicted in Fig. 2.3, which contains an ambiguous image of a dog (where the annotator can mislabel it) and an unambiguous image of a cat (where the annotator is unlikely to mislabel it).

2.1.4 Instant-dependent and semantic noise

Symmetric and asymmetric noise types are instance independent, which means that the information present in the image does not influence the labeling process. Such instance-independent noise types may be considered unrealistic since in practice, some images may be more ambiguous than others. For instance, recalling the example in Fig. 2.3, notice that the image of the dog (left) can be easily confused with the image

Figure 2.3: Example of ambiguous images of a dog (left) and an unambiguous image of a cat (right). A human annotator will typically have more difficulty to label correctly the ambiguous image on the left.

of a cat, but the image of the cat (right) is unlikely to be confused with an image of a dog. Hence, instance-dependent label noise has been proposed as another type of label noise that can better reflect the mistakes that happen in the process of labeling images.

The representation of instance-dependent label noise is usually based on an independent transition matrix (such as the ones in Fig. 2.2) per image. The instance-dependent label noise can also be represented by a labeling model that takes an image in the input and produces a label in the output, where this model is usually trained with a small clean-labeled dataset. Such representation is usually referred to as semantic label noise and an example of this model was proposed by Lee et al. [89], who trained DenseNet-100, ResNet-34 and VGG-13 using 5% and 20% of CIFAR-10 and CIFAR-100 training samples with clean labels. After training, the model was used to produce the labels of the remaining training samples. While representing a more realistic type of noise than symmetric or asymmetric noise types, the semantic noise is relatively impractical due to their dependency on a small clean training set, and the fact that the trained models

can easily overfit these small training sets.

Recently, there has been a new proposal for a model to generate synthetic instance-dependent noise utilizing semantic correlation between training samples, as described in Algorithm 1 [22]. This model associates each sample with a noisy class probability distribution vector instead of a transition matrix, where highly-correlated samples should produce similar distributions. Assuming that we have a training set $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{N}$, with images $\mathbf{x}_n \in \mathcal{X}$ and labels $\mathbf{y}_n \in \{1, ..., C\}$, the model is represented by $\mathbf{z}_n = f_\theta(\mathbf{x}_n)$, where $\mathbf{z} \in \mathcal{Z} \subset \mathbb{R}^S$ represents the extracted image feature, and $\theta \in \Theta$ is the model parameters. In Algorithm 1 [22], after pre-training the model, the noise rate for each instance is obtained by sampling from a normal distribution with mean ϵ and standard deviation $\sigma = 0.1$ and truncate the result to be in $[0, 1]$, as in $q_n \sim \mathcal{N}(\epsilon, \sigma, [0, 1])$, where $n \in \{1, ..., N\}$. Subsequently, a matrix $\mathbf{W} \in \mathbb{R}^{S \times C}$ is formed by sampling $S \times C$ times from a zero-mean one-standard deviation normal, where C is the number of classes and S is the feature size. This matrix is used for generating individual noise distribution vector for each training sample with $\mathbf{p}_n = \mathbf{z}_n \times \mathbf{W}$, where $\mathbf{p}_n \in \mathbb{R}^C$. Then, we exclude the true class from the potential label flipping choices by setting $\mathbf{p}_n(\mathbf{y}_n) = -\infty$ and distribute the noise rate q_n over the noisy labels with $\mathbf{p}_n = q_n \times \text{softmax}(\mathbf{p}_n)$, and assign the true class with the remaining probability mass with $\mathbf{p}_n(\mathbf{y}_n) = 1 - q_n$. The noisy label for the n^{th} training sample is then sampled from the categorical distribution $\mathbf{p}_n$. This algorithm is considered to be instance-dependent since training samples $(\mathbf{x}_i, \mathbf{y}_i), (\mathbf{x}_j, \mathbf{y}_j)$ with high feature embedding similarity, i.e., $\frac{\mathbf{z}_i^\top \mathbf{z}_j}{\|\mathbf{z}_i\|_2 \|\mathbf{z}_j\|_2} \approx 1$, will have equivalent flipping vectors since $\mathbf{z}_i \times \mathbf{W} \approx \mathbf{z}_j \times \mathbf{W}$.

Real-world instance-dependent label noise can also be naturally present in large-scale datasets, such as Webvision [94], Clothing1M [168], Food101 [76], Animal10N [140], and Controlled Noisy Web Labels (CNWL) [71]. These datasets represent realis-

Algorithm 1 Generation of instance-dependent noise process.

procedure NOISEGENERATION($f_\theta(.)$,$\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N\}$, ϵ, C, S)
  ▷ *$\mathcal{D}$: noisy training set* ◁
  ▷ *ϵ: noise rate* ◁
  ▷ *C: number of classes* ◁
  ▷ *S: the size of the feature vector* ◁
  Step 1: Pretrain the model $f_\theta(.)$ with $\mathcal{D}$ to obtain feature vector of training samples
  Step 2: Sample instance flip rates $\{q_n\}_{n=1}^N$ from the truncated normal distribution $\mathcal{N}(\epsilon, \sigma = 0.1, [0, 1])$;
  Step 3: Sample $\mathbf{W} \in \mathbb{R}^{S \times C}$ from the standard normal distribution $\mathcal{N}(0, 1)$;
  for $n = 1$ to N **do**
    Step 4: $\mathbf{p}_n = \mathbf{z}_n \times \mathbf{W}$
    ▷ *Generate instance dependent probability vector with size $1 \times C$* ◁
    Step 5: $\mathbf{p}_n(\mathbf{y}_n) = -\infty$
    ▷ *Exclude true label from potential flipping choices* ◁
    Step 6: $\mathbf{p}_n = q_n \times \text{softmax}(\mathbf{p}_n)$
    ▷ *Make sure the flipping likelihood to wrong classes equal to q_n* ◁
    Step 7: $\mathbf{p}_n(\mathbf{y}_n) = 1 - q_n$
    ▷ *Make sure the sum or probability flipping is 1* ◁
  Step 8: Randomly choose a noisy label from the label space according to $\mathbf{p}_n$;
  Step 9: **return** the noisy labels produced with the algorithm.

tic but challenging benchmarks because of their large size and their instance-dependent label noise that is hard to handle. Fig. 2.4 shows examples of clean and noisy-label samples from these real-world datasets.

2.2 Noisy-label learning methods

Many methods have aimed to address the noisy-label learning problems presented in Section 2.1. These methods can be categorized into the following groups: noise transition matrix [59, 125, 143], robust loss [154], [158], label cleaning [88], [182], sample weighting [131], meta learning [50], ensemble learning [114], and mixed methods [82,

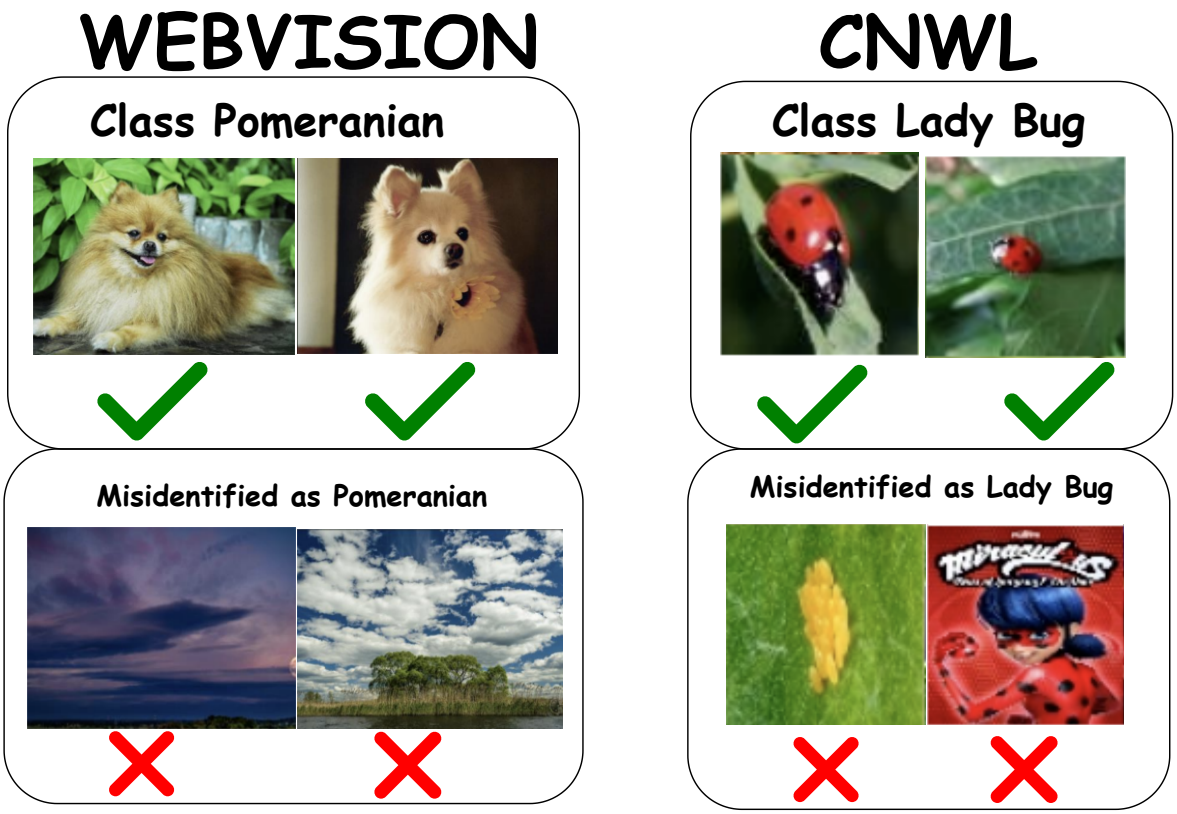


Figure 2.4: Examples of real-world Webvision and CNWL dataset containing clean and noisy labels.

189]. We provide the details of each approach below.

2.2.1 Noise transition matrix

This is one of the first methods [59, 104, 125, 143] devised to handle the class-dependent asymmetric label noise. The gist of these methods involve the estimation of the noise transition matrix, such as the ones illustrated in Fig. 2.2, used to synthesize the label noise. The idea of the noise transition matrix stems from the expansion of the

probability of the noisy label $\tilde{\mathbf{y}} \in \mathcal{Y}$ given the image $\mathbf{x} \in \mathcal{X}$, as follows:

$$\begin{aligned} p(\tilde{Y} = \tilde{\mathbf{y}}|X = \mathbf{x}) &= \sum_{\mathbf{y}\in\mathcal{Y}} p(\tilde{Y} = \tilde{\mathbf{y}}, Y = \mathbf{y}|X = \mathbf{x}) \\ &= \sum_{\mathbf{y}\in\mathcal{Y}} p(\tilde{Y} = \tilde{\mathbf{y}}|Y = \mathbf{y}, X = \mathbf{x}) p(Y = \mathbf{y}|X = \mathbf{x}) \end{aligned} \tag{2.1}$$

where X, Y and $\tilde{Y}$ denote the random variables for the image, clean label and noisy label, respectively. In (2.1), the term $p(\tilde{Y} = \tilde{\mathbf{y}}|Y = \mathbf{y}, X = \mathbf{x})$ denotes the probability transition matrix from the clean label to the noisy label, which originally is dependent on the image, but it can also be independent of the image by assuming $p(\tilde{Y} = \tilde{\mathbf{y}}|Y = \mathbf{y}) \approx p(\tilde{Y} = \tilde{\mathbf{y}}|Y = \mathbf{y}, X = \mathbf{x})$.

Many approaches have been developed to estimate such transition matrix. For example, Patrini et al. [125] estimate this matrix using a pre-trained model, Hendrycks et al. [59] calculate the transition matrix using an external clean confident set, while Sukhbaatar and Fergus [143] estimate it based on differences produced by clean and noisy data to form the transition matrices. The common assumption of those methods is the independence of label noise from the corresponding image.This is , however, not true in general since ambiguous images will have higher probability of being mis-labeled compared to images that clearly depict the samples of a particular class. Additionally, depending on the problem being studied, the transition matrix may be non-identifiable. Identifiability is a property defined by Liu [104] as the capability of the model to identify one and only one transition matrix given the training dataset, but that property depends on certain aspects of the training set, which is a topic being currently studied in many papers [95, 104, 165, 167, 174]. A previous transition matrix for noisy label work Xia et al. [166] concluded that the transition matrix demands additional assumptions in

order to be identifiable. As the result, recent transition matrix approaches are usually based on an assumption about the availability of anchor points, which is defined as data instance that has the probability of belonging to a specific class equal to one Liu and Tao [103]. This assumption is reasonable in certain scenarios and has been exploited by Liu and Tao [103]; Patrini et al. [125]). Unfortunately, this assumption does not hold in some cases, which could lead to a incorrect transition matrix and degrade the classifier's performance Xia et al. [166]. Accordingly, this problem has inspired the born of many transition matrix estimation framework without the needs of anchor points (Xia et al. [166]; Liu and Guo [105]; Xu et al. [170]; Zhu et al. [206]). For instance, in the work of Zhu et al. [206], the author proposes an assumption that each noisy training sample will share similar ground-truth label with its two other nearest neighbor samples (2-NN), and estimate the transition matrix based on that. Unfortunately, Zhu et al. [206] have also provided empirical evidence that this assumption is not true in practice, showing an experiment that indicates that only around 60-78% of noisy training samples satisfy this condition.

2.2.2 Robust loss

One of the common approaches to address noisy-label problem is to make the loss function robust to noisy labels by reducing the penalization for samples where the label from the model does not agree with the label from the annotation. This will prevent the training algorithm to overfit the noisy labels from the training set. In Manwani and Sastry [111], it is shown that the 0-1 loss is more noise-tolerant than other commonly used classification losses. The mean absolute error (MAE) loss is shown to be more robust to noisy labels [41], as it treats all samples equally, but depending on the specific

type of noise, it can lead to underfitting. The loss function in Ziyin et al. [207] works by reducing the weight or removing potential noisy samples. Another example is Ma et al. [109], which introduces a methodology to normalize any loss function to reduces overfitting, but risks underfitting the training data.

Unfortunately, the downside of these approaches is the slow convergence and underfitting. In particular, a slow-converged-but-robust loss function, while exhibiting resilience against noisy labels, might lead to underfitting, while a fast-converged-but-robust loss function will likely overfit noisy labels. Also, any mechanism to dynamically update the type of loss over time is usually ad-hoc for specific datasets or noise types and often demands an extensive hyperparameter-tuning to achieve reasonable results. For instance, DivideMix [92] uses two types of loss, namely a fast-converged cross-entropy loss for samples classified as clean and a robust mean squared error loss for samples classified as noisy, where the robust loss Mean Square Error (MSE) loss is weighted by a pre-defined hyper-parameter. This hyper-parameter needs to be adapted for different benchmarks, which means that it assumes knowledge of the type and rate of noisy label affecting the dataset. Such assumption is arguably too strong, and more automated loss function weighting should be investigated.

2.2.3 Sample weighting

Sample weighting consists of mechanisms that aim to down-weight the loss from noisy samples and up-weight the loss from clean samples [131, 139, 197]. These methods require some assumptions about the characteristic of noisy samples, which are often based on confidence score or informativeness. The most relevant sample-weighting methods can be listed as follows. Curriculum-based methods define a weight to each

sample based on an estimate of sample complexity [46] or on an estimate of outlier probability [172]. A similar approach in Wang et al. [157] minimizes the influence of noisy labels by down-weighting training samples near the prediction boundary. gIn light of weighting training samples, CleanNet [90] takes advantages of a reference subset and estimates the weight of training samples based on their cosine similarity distance in feature map with reference samples, while Harutyunyan et al. [54] uses the gradient from the last layer of the network to estimate the sample weight. These methods can, however, only detect clean-label samples, which can be detrimental to the training process since clean samples are usually considered to be uninformative (i.e., the cleaner the sample, the more confident the model is about it, and the less informative it is for model training).

2.2.4 Sample selection and noise identification

Sample selection is one of popular approaches to handle noisy label problems. For instance, MentorNet [70] uses two networks, namely a teacher and a student networks, with the latter being trained from samples that are likely to be clean. This model relies on curriculum learning to schedule the selection of clean samples for training. SELF [119] identifies noisy samples using a filtering mechanism based on a model ensemble, but it requires a clean subset to achieve a good performance. The influential paper by Arpit et al. [6] argues that deep neural networks tend to learn clean-label samples faster before overfitting the noisy samples, leading to lower loss for clean samples at the beginning of the training [19, 51]. This observation is known as the "small loss trick", which assumes that clean and noisy samples have significantly different distributions for their classification losses, particularly during the first training iterations (i.e., the warm-

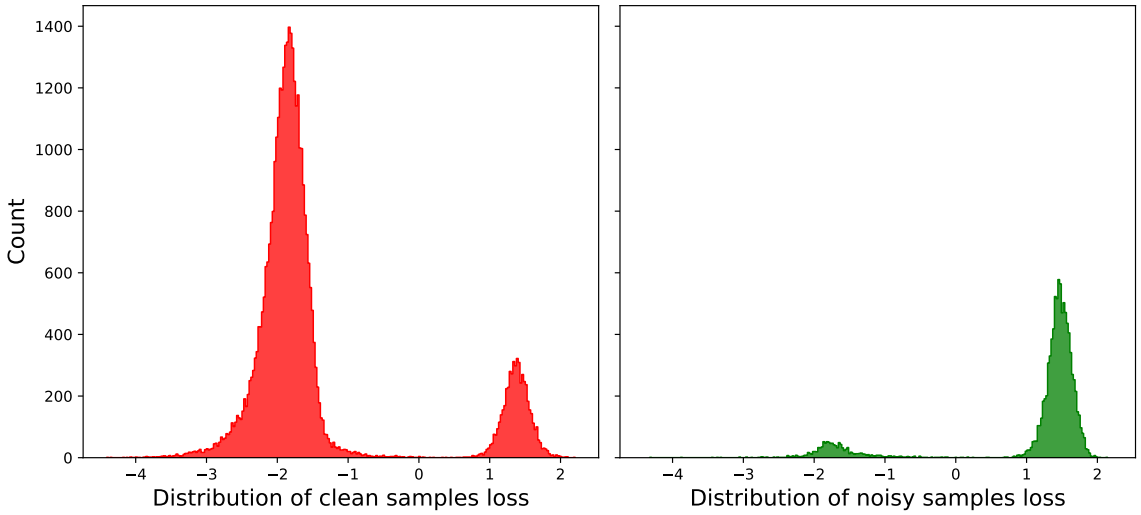


Figure 2.5: Distribution of natural logarithm of the cross-entropy classification loss values for clean-label samples (left) and noisy-label samples (right) after 30 epochs of vanilla training with noisy-label data using a Resnet18 model on CIFAR-10 dataset with 40% asymmetric noise.

up training stage). Despite the lack of theoretical guarantee for this observation, it has been adopted by many works [26, 70, 92, 200], in which clean samples are estimated based on their loss after a few epochs of vanilla training with noisy-label data. Fig. 2.5 displays an empirical distribution of loss values of the clean and noisy-label samples, demonstrating this assumption on a synthetic label noise problem (CIFAR-10 dataset with 40 % asymmetric noise) using a pre-trained DivideMix model [92] with a Resnet18 model. Under such heavy asymmetric noise, the difference of the loss distributions from clean-label and noisy-label samples is significant. For instance, note that the majority of noisy-label losses range from 2 to 6, while most clean-label samples have their losses close to 0.

In order to detect/separate noisy-label data, Sanchez et al. [135] use a two-component Beta Mixture Model (BMM) and train with the max-normalized loss to classify clean and noisy-label samples. However, according to Li et al. [92], BMM can produce unde-

sirable flat distributions and can also fail for the asymmetric noise case. Also according to Li et al. [92], Gaussian Mixture Model (GMM)[126] is a better choice to distinguish clean and noisy samples due to its flexibility in the sharpness of distribution.

Besides the small loss trick, several other attempts to differentiate noisy-label data from clean-label ones have been introduced. For instance, Kim et al. [80] proposed a mechanism to identify noisy-label samples based on the first eigenvector computed from the feature vectors of each class. This approach exploits two assumptions: 1) the prevalence of clean-label samples in each class, and 2) the first eigenvector from each class' deep feature representations summarizes the most important characteristics of that class. Another prominent noisy-label sample identification was proposed by Kim et al. [79], which takes into account the inherent correlation of each training sample with its neighboring samples in the feature space. This method estimates the pseudo label for each training sample from the prediction logits of a significant number of neighboring samples, which are used by a MixUp operation [187] between the target sample and its neighbors.

Once the noisy-label sample identification process has finished, the initial sample selection approaches were designed to keep the clean-label samples for training, leaving the remaining noisy-label samples to be trained with regularization loss [197] or robust losses [26, 92]. Fig. 2.6 shows a traditional sample selection method, in which a GMM is utilized to cluster training data into a pseudo clean and a pseudo noisy sets. After that, unsupervised or semi-supervised learning [92, 200] is adopted to train with the pseudo noisy set, while the pseudo clean set is employed to train the network weight with cross entropy loss.

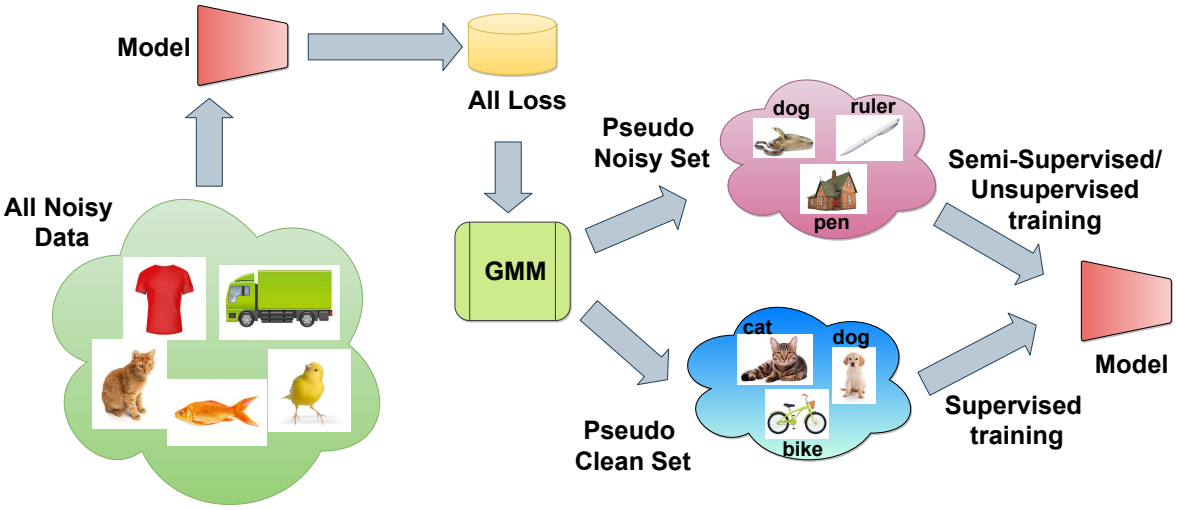


Figure 2.6: A generic framework for sample selection with GMM.

2.2.5 Pseudo labeling

Pseudo labeling is technique for automatically generating one-hot pseudo label vector for each training sample to guide the learning process. One of the initial approaches for pseudo labeling is based on a simple network prediction, taking advantage of the inherent robustness of deep neural networks at early training stages. It has been verified empirically by Liu et al. [102] that deep learning models can learn the correct label of noisy-label samples at the start of training phase, and only gradually get contaminated and overfit to noisy-label signal in the subsequent phase. Hence, early studies proposed to handle noisy-label learning exploit this attribute of deep neural networks and predict the clean pseudo labels using a warmed-up model with early stopping. A recent pseudo labeling approach proposed by Shi et al. [138] uses the correlation between noisy-label samples and their clean neighbors to approximate the samples' clean labels. Another example of a prominent pseudo labeling work based on the use of neighboring information is a recently published work by Iscen et al.

[67] that presents a graph-based label propagation [67]. In this paper, the authors represent noisy-label training samples in training set as nodes of a similarity graph and the graph information is periodically updated to produce reliable pseudo labels. While the methods above have shown strong results, it has been pointed out by Arazo et al. [4] that naive pseudo labeling can degrade the model's performance, with a problem known as confirmation bias, where prediction errors are memorized by the network.

In meta learning, this issue is noticeable due to the usage of a single model by traditional meta learning framework for noisy label, which we will demonstrate with empirical evidences in Chapter 3. Moreover, label propagation is fairly expensive given the necessity of the knowledge of deep feature representation of the whole noisy-label training set. Each time we need to update the label, extra computational expense is incurred by label propagation due to model inference for the whole dataset. Hence, in the proposed methodology presented in Chapter 3, I employ pseudo labeling indirectly by identifying noise-contaminated samples instead of letting the model fit the generated pseudo-label. I also only use fairly simple label estimation techniques to avoid the expensive computation in the graph propagation methods.

2.2.6 Data augmentation for noisy-label learning

Data Augmentation is a solution developed to tackle the problem of limited data availability for training deep neural networks. This methodology consists of a suite of techniques that amass a significant amount of training data by exploiting existing available dataset. Popular image augmentation techniques includes: geometric transformations comprising simple image transformations such as flipping, cropping, rotation, and translation; color space augmentations referring to methods that systematically ma-

nipulate image color channels; kernel filters to sharpen or blur images; image mixing to generate new training data from linear combinations of existing samples; and randomly erasing parts of image.

Besides the model-independent augmentation techniques above, a few augmentation methods that make use the deep neural networks have also been proposed recently, such as: feature augmentation methods that manipulate the lower-dimension deep learning embeddings from intermediate network layers [34, 85, 163], adversarial training which exposes the weakness of the target network by identifying the minimum possible noise injection needed to cause a misclassification with high confidence [116, 142, 184], and data generation that produces new data samples which retain similar characteristics of the target classes [38, 98, 129]. The combination of augmentation methods has been introduced to solve noisy-label and semi-supervised problems. The proposed framework presented in Chapter 3 exploits data augmentation in the training of our meta learning approach.

2.2.7 Meta learning

Another noisy-label technique that has achieved impressive performance is based on meta learning. In recent years, meta learning has become an important topic in the machine learning community. This approach relies on a meta-parameter that automatically weights training samples for training the model. Meta learning learns to learn using a meta model that looks one or multiple steps ahead and automatically infers the optimal rule for loss adjustment. Loss re-weighting with meta learning aims to assign smaller weights to the noisy samples and greater weights to clean samples.

In previous meta learning approaches that automatically re-weight training samples

[1, 2, 131, 139, 149, 197], we can find a common set up that requires 2 data subsets: a small subset of data with clean labels and a large subset of noisy-label samples. One of the first meta learning papers in noisy-label learning is *Learning to Reweight* (L2W) [131], where the meta-parameter is estimated using a clean-label validation set. This method re-weights the training samples, where the weights are updated based on the gradient information from the loss that is minimized with respect to the clean validation set. These meta-parameters are then used to weight training samples' during the model training. Fig. 2.7 shows a diagram and explanation for L2W framework - one of the initial meta learning for noisy label approaches Ren et al. [131]. For each mini-batch randomly sampled from the whole noisy dataset, this model estimates the expected one-step-ahead meta model being trained with the cross entropy loss over the mini-batch (the contribution of each noisy sample in the mini-batch is determined by a meta weight, which is initially initialized uniformly). Subsequently, the loss of the meta model over the validation set is estimated, and then minimize by taking derivative with respect to the meta weight of the noisy mini-batch. Finally, the framework updates the target model with a weight cross entropy loss using the noisy mini-batch and their renovated meta weight.

Similarly, Zhang et al. [197] is another meta learning approach that relies on a clean validation set, but instead of just weighting training samples, it also relabels the training samples, allowing it to achieve state of the art performances. The work by Shu et al. [139] also uses a meta learning approach with a clean validation set, but they use a multi-layer perceptron to learn a sample-weighting function. Additionally, L2LWS [32] and CWS [31] rely on a target DNN and a meta DNN. The target DNN receives guidance from the meta DNN pre-trained on a small clean validation dataset to re-weight training samples.

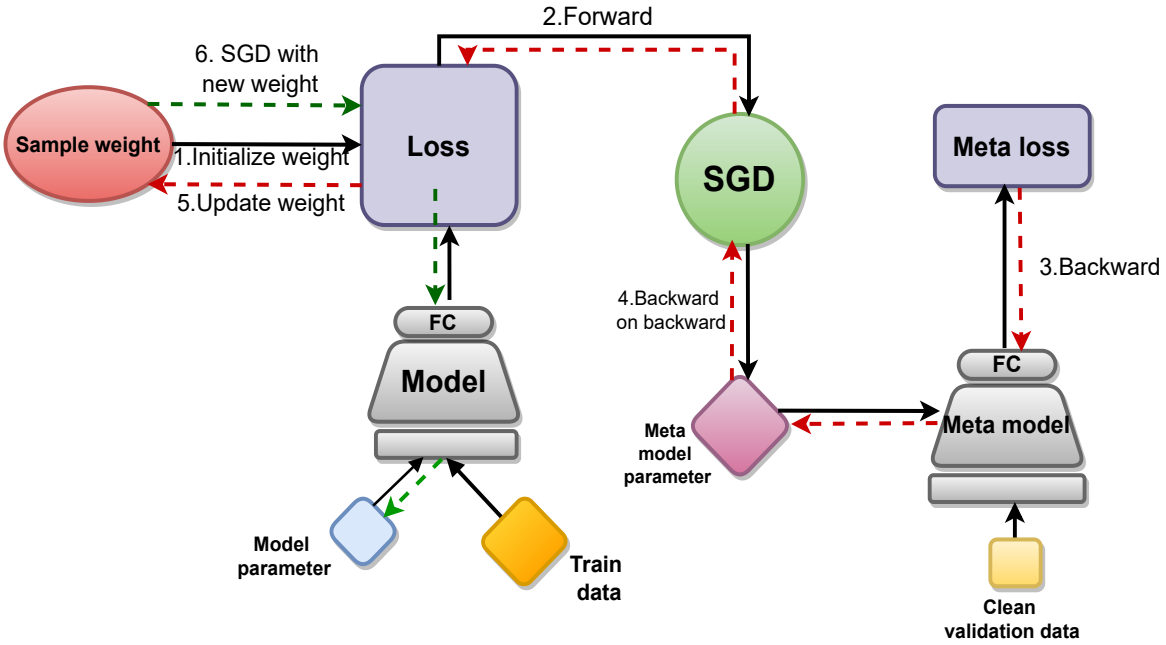


Figure 2.7: The L2W meta learning diagram contains a model, which is the classifier, and a meta-model, represented by the weights for the training samples. The training sample weights are initialized to have uniform weights within a batch (1). The model is then trained with the weighted training data using a weighted cross entropy loss (2). Next, these weights are estimated by minimizing the meta-loss using a clean validation set (3), which requires a second-order gradient descent (4). After updating the training samples weights (5), we re-train the model with a weighted cross entropy loss (6). In the figure, SGD means stochastic gradient descent.

The main issue with the methods mentioned in this section is the need of a clean validation set, which limits the types of datasets that can be handled (e.g., clean-label validation sets become more complicated to obtain as their number of classes increases). In practice, because of the lack of expert labels and high annotation costs, the clean set is much smaller compared to the noisy set. Despite that, it can still be expensive to manually create this clean set for more complex problems with thousands or millions of classes. A recent work that removes the need of a clean validation set is the FSR model [196] that automatically selects a validation set based on the confidence

Table 2.1: Test accuracy (%) of meta learning approaches on CIFAR datasets under symmetric noise at several rates (from 0.2 to 0.8). Those masked with (T) are methods that require a clean validation set.

Method	Clean val	CIFAR10			CIFAR100		
		0.2	**0.4**	**0.8**	**0.2**	**0.4**	**0.8**
L2R^{T}	Yes	90.0 ± 0.4	86.9 ± 0.2	73.0 ± 0.8	67.1 ± 0.1	61.3 ± 2.0	35.1 ± 1.2
MWN^{T}	Yes	90.3 ± 0.6	87.5 ± 0.2	-	64.2 ± 0.3	58.6 ± 0.5	-
GDW^{T}	Yes	-	88.1 ± 0.4	-	-	59.8 ±1.6	-
MentorNet^{T}	Yes	92.0 ± 0.0	89.0 ± 0.0	49.0 ± 0.0	73.0 ± 0.0	68.0 ± 0.0	35.0 ± 0.0
Distill^{T}	Yes	96.2 ± 0.2	95.9 ± 0.2	93.7 ± 0.5	81.2 ± 0.7	80.2 ± 0.3	**75.5 ± 0.2**
FaMUS	No	-	95.3 ± 0.2	-	-	76.0 ± 0.2	-
FSR	No	95.1 ± 0.1	93.7 ± 0.1	82.8 ± 0.3	78.7 ± 0.2	74.2 ± 0.4	46.7 ± 0.8

gain obtained for training samples after each training epoch. However, this approach has much lower accuracy compared to meta learning approaches that require a clean validation set, especially in high noise rate scenarios. Table 2.1 compares the results of previous meta learning methods over CIFAR datasets under various rates of symmetric noise. The results in Table 2.1 shows that meta learning methods that do not rely on a clean validation set (FSR [196] and FAMUS [171]) have inferior results compared to methods that exploit a clean validation set Zhang et al. [197].

The inferior results by FSR [196] and FAMUS [171] are quite obvious due to the prevalence of noise in their validation sets, which lead to confirmation bias and model overfitting. Even though methods that depend on a clean validation set [131, 197] show better performance, they have an severe weakness, which is that the clean subset will not change for the whole training process. As the models get more accurate, the increase in confidence for the validation set samples makes the set less informative for the noisy re-weighting process. That arises a need to design a mechanism to update the validation set periodically to further improve the training.

2.2.8 Mixed methods

In addition to the methods presented in Section 2.2-2.2.7, there are several approaches that rely on a combination of several techniques. One of the most influential methods proposed in the last couple of years is DivideMix [92], which mixes semi-supervised learning, co-teaching and sample selection based on the small-loss trick explained in Section 2.2.4. In another interesting study Kim et al. [82], instead of forcing the predicted labels to be closer to the potentially noisy training labels, the predicted labels are forced to be far away from classes that are likely to be incorrect to make the training more robust to noisy samples. These two methods , however, require longer training times and may be negatively impacted by a large number of classes. Another crucial method that is worth mentioning is MixUp [187], which is a data augmentation technique that generates and trains the model using linear combinations of images and labels from the noisy training set. Empirical results by the SOTA methods (DivideMix [92], PropMix[26]) demonstrate that MixUp is adaptable to many models and robust to any type of label noise. Thanks to improvements provided by MixUp [187], recent noisy label learning approaches started to incorporate this method as an important part of their algorithms. For example, Zhang et al. [197] propose a combination of meta learning and MixUp, which results in a tremendous performance boost in the majority of noisy label benchmarks.

The strengths and weaknesses of each noisy label learning method are summarized in Table 2.2 based on adaptability (how easily this method can be integrated into new frameworks), competitiveness (how accurate the results are in recent benchmarks), practicality (how easily it can be implemented), and if it needs clean data. The table shows that meta learning ticks several of the boxes, but its dependence on a clean

Table 2.2: Advantages and disadvantages of noisy-label learning approaches. **Adaptability** denotes the flexibility of the methods that can be integrated into any framework, **Competitive with SOTA** denotes if the approach (by itself) can produce state-of-the-art results in current benchmarks, **Practicality** represents if the method is simple and easy to implement, **Clean validation data** is True if the methods require extra data.

Method	Examples	Adaptability	Competitive with SOTA	Practicality	Clean data
Transition Matrix	[59, 125, 143]	✓		✓	
Robust Loss	[154, 158, 159]	✓		✓	
Sample Weighting	[70, 131, 139, 196]	✓		✓	✓
Sample Selection	[26, 92, 200]	✓		✓	
Meta Learning	[131, 139, 197]	✓	✓	✓	✓
Mixed	[26, 92]		✓		

validation set is a disadvantage that needs to be addressed.

2.3 Imbalanced learning

Imbalance learning is one of the most challenging problems that is widely present in real-world datasets. This problem is characterized by a long-tailed class distribution, where a small number of majority classes contain a large number of training samples, and a large number of minority classes contain a small number of training samples [193]. Imbalance learning can substantially degrade the performance of trained deep neural networks, which tend to become biased toward the majority classes, while having low accuracy on the minority classes. According to Zhang et al. [192], imbalance learning is characterized by the imbalance ratio that is computed with the ratio between the number of samples in first majority class and the last minority class. To address this problem, many imbalanced learning methods have been proposed, which can be categorized as follows (see Fig. 2.8): class re-balancing [63, 72, 152, 156], information

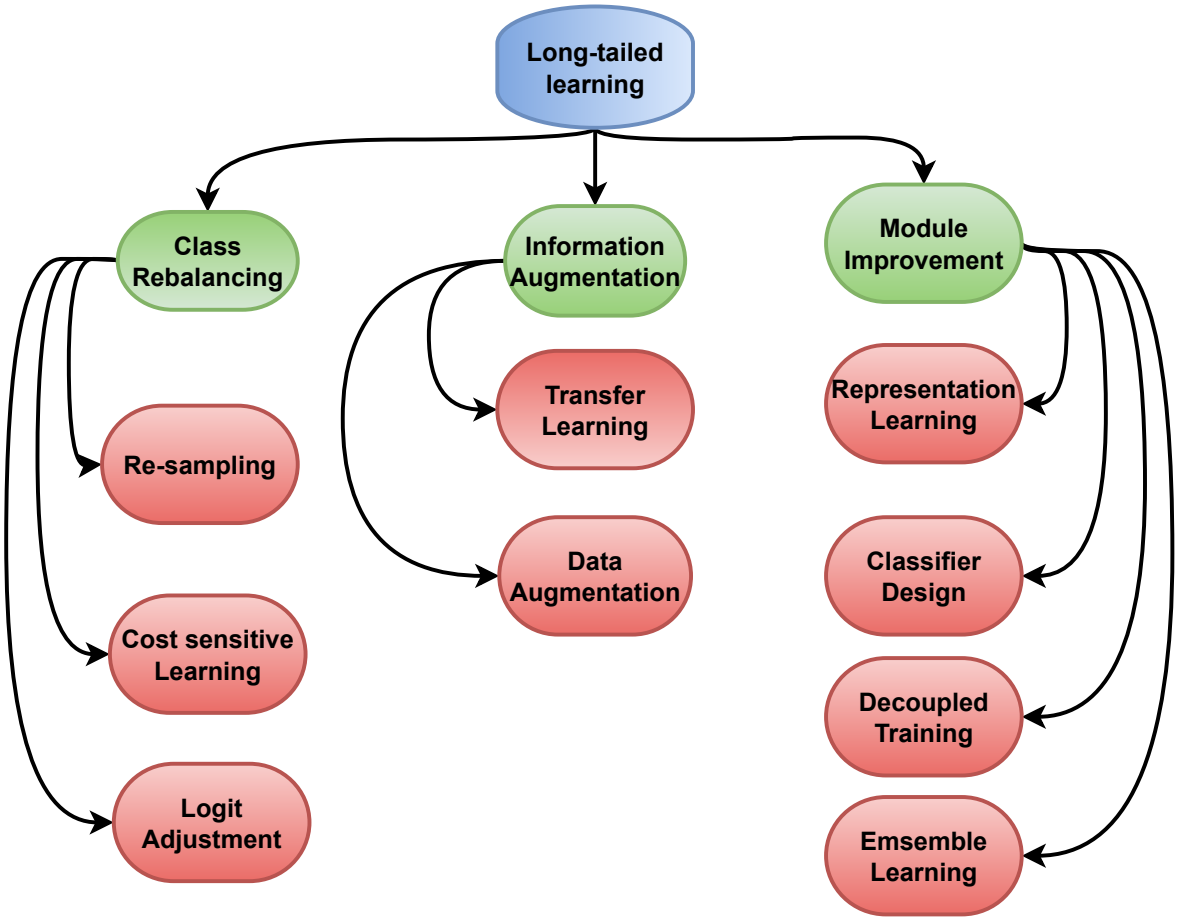


Figure 2.8: Summary of imbalance learning methods.

augmentation [42, 44, 62, 93, 161, 162, 173, 185], and module improvement [27, 35, 35, 45, 72, 73, 73, 99, 146, 203]. The details of each approach is described in details in the sub-sections below.

2.3.1 Class re-balancing

Class re-balancing methods aim to "repair" the imbalance of training samples from different classes during the training. Such objective can be achieved in three ways:

- Re-sampling: this approach randomly over-samples (ROS) the minority classes

and randomly under-sampling (RUS) the majority classes. Some prominent examples of re-sampling for imbalanced learning were proposed by Kang et al. [72], Zhang and Pfister [196]. Meta learning is also adopted as a part of this branch of methodologies. For instance, Ren et al. [130] proposed the balanced meta-softmax framework based on a meta learning mechanism that approximates the ideal sampling rate for each class. Specifically, their methodology, which demands the usage of an extra balanced validation set, involves of a bi-level optimization policy to explore the best meta parameter that represents sampling ratio for each class, and uses this meta parameter for cross entropy training. Nevertheless, for datasets with large imbalance ratio, ROS can overfit the minority classes and RUS can underfit the majority classes, which are problems that have been mitigated by recently proposed re-sampling approaches, such as class-balanced re-sampling [152, 156] and scheme-oriented sampling [78, 203].

- Cost-sensitive learning: these approaches consist of adjusting the loss function according to the class imbalance ratio of the training set. Recent studies follow two strategies, including class-level re-weighting [29, 97] and class-level re-margining [28, 39]. Intuitively, class-level re-weighting approaches weight the training sample's contribution in a class-wise manner, in which the most common class-level re-weighting strategies [29, 97] simply use label frequencies of training samples as the contribution factor in the final loss. In contrast, class-level re-margining [28, 39] aim to tackle the class imbalance problem by modifying the minimal distance between the deep representative features of each class with respect to the frequency of such classes in the dataset. For instance, label-distribution-aware margin (LDAM) [15] framework attempts to enforce minority class representa-

tion vectors to have larger margins with respect to the classification boundary. However, these loss modification approaches are usually class-dependent. Hence, these methods may underestimate or overestimate the imbalance ratio when dealing with noisy-label and imbalanced learning.

- Logit adjustment: this strategy [63, 148] targets the artificial adjustment of logits based on class sample distributions with the goal of minimizing the average per-class error [192]. Unfortunately, similar to cost-sensitive learning, logit adjustment also needs a clean dataset to reliably estimate the class imbalance ratio.

2.3.2 Information augmentation

Information augmentation adds extra information to train the model in order to improve the performance for imbalanced learning problems. Following Zhang et al. [192], methods following this idea can be categorized into:

- Transfer learning: approaches that transfer the information from a source domain to enhance model performance on a target domain. In the context of imbalanced learning, there are four different strategies for transfer learning, including self-training [161, 162], model pre-training [42, 173], head-to-tail knowledge transfer [24, 151] and knowledge distillation [44, 62].

- Data augmentation: these methods explore augmentation techniques to generate a large amount of extra data to boost the size and quality of input for model training. Currently among previous augmentation-based works that attempt to combat class imbalanced problem, there are two prominent strategies, namely: transfer-based augmentation [58, 161, 186] and the classic non-transfer augmen-

tation [93, 185]. Transfer-based augmentation targets the minimization of differences between the majority and minority classes by augmenting model training, since there exist traits shown by the majority classes, but may not be shown by the minority classes. For instance, FTL [178] has shown that majority class samples have bigger intra-class variance compared to minority class samples. Inspired by this observation, they address the imbalanced-learning problem by minimizing such variance differences between the majority and minority classes, leading to significantly better performance for the minority classes. Non-transfer augmentation simply utilizes a combination of traditional image augmentation techniques to address imbalanced-learning problems. According to Zhong et al. [202], MixUp [187] has proven to be able to alleviate the over-confidence problem caused by imbalanced-learning. Additionally, according to their empirical results, representation learning for imbalanced data also benefits greatly from MixUp.

Overall, both transfer learning and data augmentation have demonstrated impressive results for imbalanced learning problems. However, a weakness of this approach is that not only it requires considerable extra computational cost but also complicates the framework. In fact, transfer learning requires either high cost for self-training [58, 161], sensitive hyper-parameter tuning, or complicated framework design [56, 60] for knowledge transfer and knowledge distillation.

2.3.3 Module improvement

The last category of imbalanced learning methods is called module improvement, which can be sub-divided into four approaches:

- Representation learning: methods that aim to improve the feature representa-

tion of training samples [27, 35] with metric learning [35, 73], sequential training [123, 201], prototype learning [106, 205], and transfer learning [100, 173]. In general, such representations are learned by approximating the representations of training samples belonging to the same class, and separating the representations of samples from different classes. Additionally, according to [200], this family of methods has proven to be effective for the noisy-label learning problem as well. The main drawback of this approach is the overhead for learning the representation.

- Classifier design: new classifiers have been proposed for imbalanced learning, such as the causal classifier [146] that aims to keep the positive causal effect (defined as the beneficial factor contributing to the stabilization of gradients and model convergence acceleration), while eliminating the negative causal effect (i.e., the accumulated gradient from imbalanced learning that degrade the model's performance for the minority classes). Another example for this family of methods is the GIST classifier [99] which seeks to bridge the gap between the geometric representations of majority and minorities classes.

- Decoupled training: this technique takes advantage of both classifier design and representation learning by optimizing the representation of training samples, as well as classifier design [72, 74]. This method assumes that the balance in the feature representation space makes significant contribution to combat imbalanced data. Hence, a k-positive contrastive loss is proposed and integrated to the framework to train the sample feature space to be more class-discriminative, leading to a better classification performance [72, 73].

- Ensemble learning: these methods systematically attempt to produce and gener-

ate a combination of multiple models to create a robust framework against imbalanced learning problems. BBN [203] consists of a 2-branch model, taking advantages of a classic target branch and a supporting branch for data re-balancing that mitigates the negative influence of imbalanced learning. Inspired by BBN, LTML [45] comprises another bilateral-branch model designed for an imbalanced multi-label classification problem, which maintains the logit consistency between the predictions from the two branches.

2.4 Active learning and its application for meta learning

Active learning is a technique developed to allow the training of models using a small training set, where the model is originally trained with few samples, and at each iteration, the algorithm selects the most informative sample to be labeled for the next round of training [132, 136]. Overall, active learning approaches can be divided into three categories: membership query synthesis [3, 83], stream-based selective sampling [30, 117], and pool-based active selection [55, 91]. Membership query synthesis comprises methods where the models can request the labeling of any unlabeled sample, including the sample generated by the model, stream-based selective sampling makes independent decisions to label each sample sequentially selected from the dataset, and pool-based active selection chooses the best sample to label based on some criterion from a small pool of instances that has been randomly extracted from the dataset. Fig. 2.9 illustrates the differences and general ideas of each active learning selection mechanism.

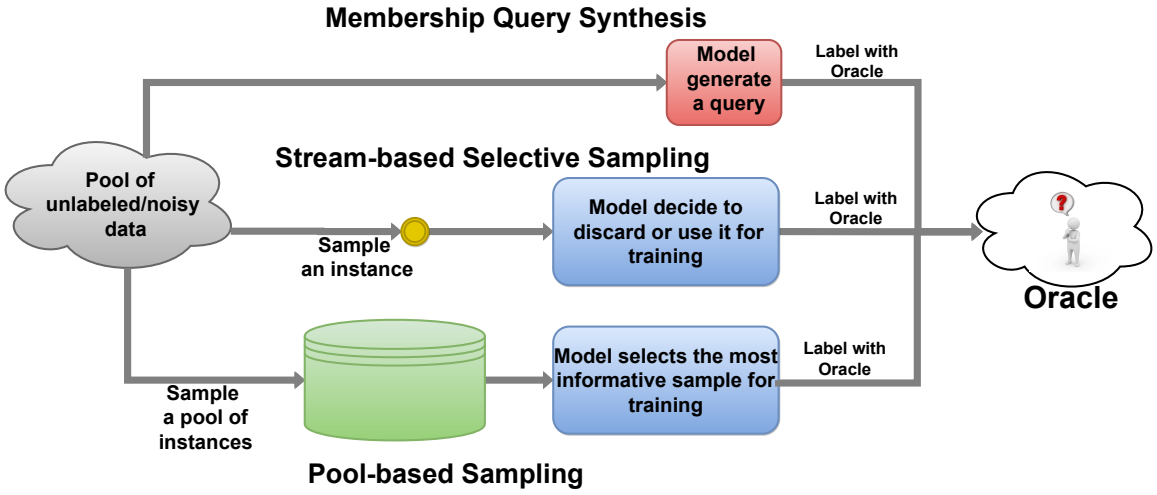


Figure 2.9: The differences between three main categories of active learning.

There have been significant efforts by the community for the development of active sample selection methods. Some noteworthy approaches are uncertainty-based methods [9, 91, 137] which select samples that the model is least likely to be certain, diversity-based approaches [12, 47], which choose samples to maximize diversity measures, and expected model change strategies [40, 133] that select samples that produce the most significant update to model weight. In addition, many works have also attempted to find a balance between uncertainty and diversity of samples [7, 177], resulting in hybrid strategies.

There have been many attempts to incorporate active learning with deep learning to reduce annotation cost, as deep learning models are known to be data hungry. A key difference between deep active learning and active learning approaches lies in the batch based sampling query. Classic active learning approaches select informative sample one by one, e.g., [65] uses a Bayesian network to determine the complexity of samples. However, this leads to highly informative but similar samples, leading to a sub-optimal batch informativeness. Hence, in [84], the authors propose BatchBALD as an extension

from [65] to maximize informativeness of each batch by removing duplicate information within samples. It is important to note that the sample selection that we used to include samples in the validation set suffers from a similar problem, where samples with high informativeness tend to be similar to one another, so in our formulation Eq. (3.9), I also study the diversity of informative samples.

For meta learning methods, I am not aware of previous attempts that try to use active selection techniques to automatically build validation sets based on label cleanliness and sample informativeness. Unlike traditional active learning methodologies that rely on a pre-determined objective function representing the informativeness of training samples, meta learning has a dynamic definition of validation set utility that depends on label cleanliness and sample informativeness at the current meta-training stage. As there is no prior works exploring beneficial characteristics for the meta validation samples, it is even harder to identify mutual and redundant characteristics between samples from the same validation set like BatchBALD. Moreover, the validation set of meta learning algorithms is required to be balanced and fairly small, which can challenge even more our active selection approach. All of these hurdles are the main reasons why the active selection of validation samples for meta learning is still an open-research problem.

2.5 Self-supervised learning

Self-supervised learning is a family of methodologies designed to provide good model initialization [69, 200]. These methods use a massive amount of unlabeled training data to pre-train large deep learning models that can then be effectively transferred to downstream tasks [20, 21, 57]. Self-supervised learning methods can be classified

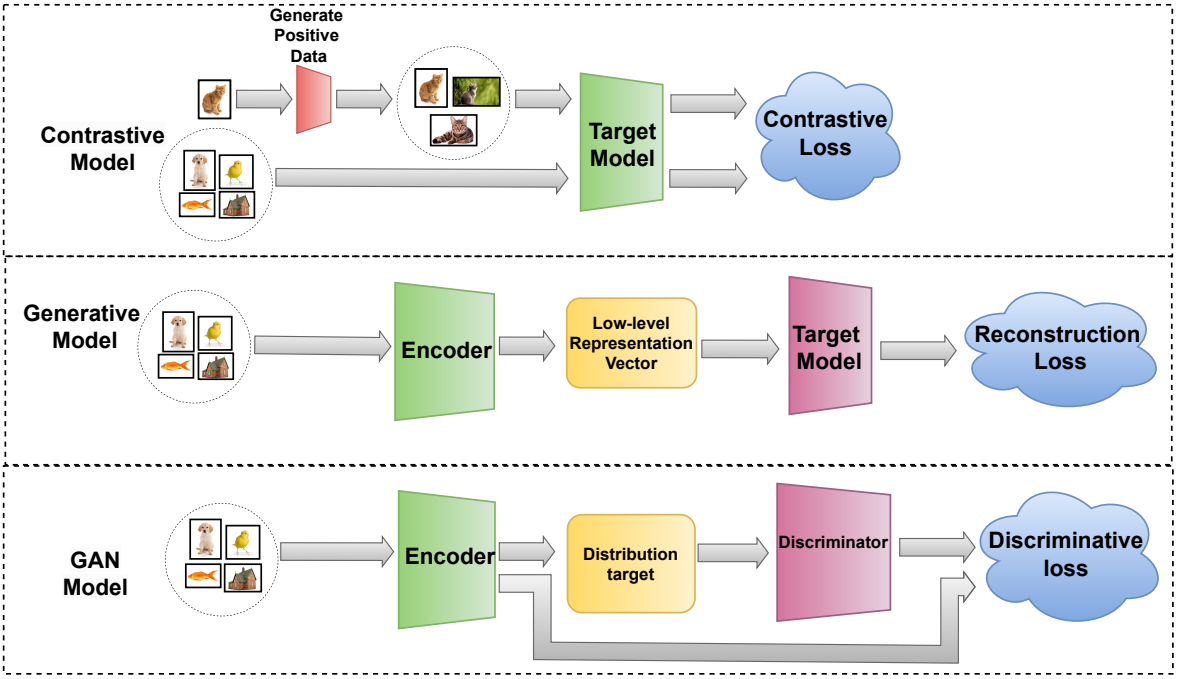


Figure 2.10: Summary of self-supervised techniques.

into three dominant categories: generative learning, contrastive learning and combined methods that explore adversarial learning. Generative learning deploys an encoder and a decoder, in which given an input image, the encoder outputs a representation vector that is used by the decoder to reconstruct the input image [134, 150, 169]. Contrastive learning uses an encoder to build a representation from an input image and formulate a similarity measure that is maximized for representations of the different views of the same image and minimized for different images [49, 200]. The hybrid generative-contrastive methods uses a discriminator to minimize the discrepancy between original samples and the generated fake samples from a trained encoder-decoder model [5, 43, 128]. Fig. 2.10 shows a summary of the three self-supervised learning techniques in Fig. 2.10.

Among these methods, contrastive learning has been adopted recently to address the

noisy-label problem and has shown promising results. Such robustness to noisy labels is achieved with the learning of representations that are less prone to effectively overfit to noisy-label samples. Some prominent methods that utilized contrastive learning for noisy-label learning are C2D[200] and MOIT [122].

2.6 Conclusion

This chapter has discussed the noisy label and imbalanced-learning problems in detail, highlighting the advantages and disadvantages of currently available approaches for these two problems. In particular, I have reviewed meta learning, pseudo labeling, self-supervised learning, data-augmentation, active learning and noise identification methods. I have analyzed the current limitations of these approaches and discussed the current knowledge gaps in the field. I have also motivated my proposed method to address these limitations and gaps in current knowledge. The literature review of several methods that are closely related to our approach and that are important in the development of the proposed approach is also included in this chapter. In the next chapter we present the main contribution of this thesis, which is a method that defines a new utility criteria to automatically build the validation set in meta learning methods without resorting to any manual labeling.

Chapter 3

Validation Set Utility Maximization in Noisy-label meta learning

In this chapter I present the main contribution of this thesis. Among previously proposed methods to address the problem of imbalanced learning and noisy-label learning, meta learning has proven to be one of the most effective approaches. However, the conventional meta learning usually requires a clean validation set of balanced-distributed samples, which relies on manual labeling process. Not only this process is expensive, it may also be infeasible in many scenarios due to data scarcity or security problems. Additionally, a randomly selected validation set not only scales poorly with the number of classes, but may be sub-optimal for meta learning. In this Chapter, I present a novel heuristic to optimize the utility of the clean validation set for the meta learning framework. The heuristic is formulated based on an analysis of the meta learning optimization and consists of three factors: sample informativeness, data cleanliness and balanced distribution. I achieve significant boost in performance over previous meta learning works and other imbalanced learning and noisy-label learning methods

on various benchmarks.

3.1 Introduction

In Chapter 2, I have discussed the noisy-label learning and imbalanced learning problems. Overall, the proposals to address these problems can be divided into several categories: robust loss functions [154, 159], label cleaning [68, 183], meta learning [53, 131], ensemble learning [113], and other methods [81, 190]. Among these approaches, meta learning has shown state-of-the-art performance when using training data that is heavily contaminated with label noise. meta learning is a bi-level optimization methodology that traditionally requires a clean and balanced validation set to optimize the meta parameters. The first attempts to formulate meta learning for noisy-label learning included a validation set that consists of randomly selected samples. Such validation set could lead to potentially sub-optimal performance [131, 139, 197] because that set is not guaranteed to be informative and may be too small for meta learning. As a result, this issue has motivated the design of ad-hoc methods to automatically build good validation set [171, 196]. However, these approaches have not demonstrated competitive results, especially in high noise-rate scenarios. A reasonable explanation for this issue is related to the limitations of their proposed heuristics [171, 196], which takes into account balanced distribution and label cleanliness but ignores the informativeness of the validation samples.

I propose a new imbalanced noisy-label meta learning (INOLML) method that automatically builds a validation set with high utility for the meta learning optimization. Such high utility validation set is characterized by samples with high informativeness, high probability of having clean labels, and well-balanced class distribution. The es-

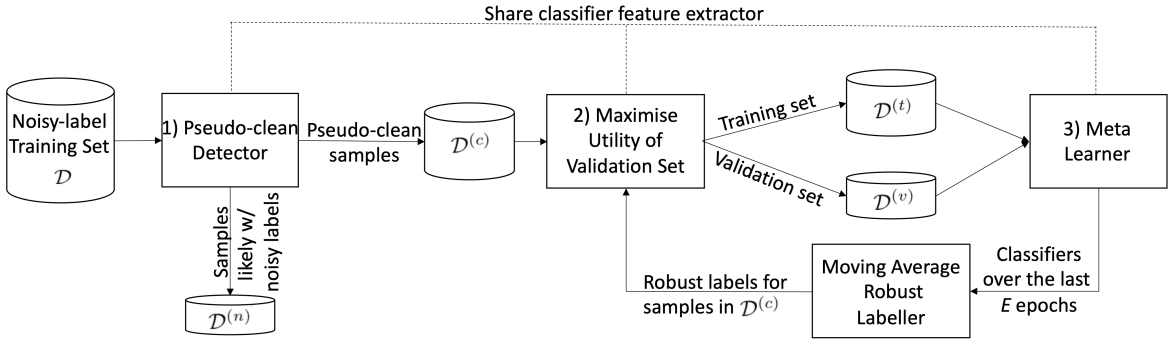


Figure 3.1: A sketch that illustrates the phases of the INOLML algorithm: 1) filtering of the initial noisy training set $\mathcal{D}$ to divide it into two disjoint sets, including a pseudo clean set $\mathcal{D}^{(c)}$ and a pseudo noisy set $\mathcal{D}^{(n)}$; 2) building the training set $\mathcal{D}^{(t)}$ and the validation set $\mathcal{D}^{(v)}$, extracted from $\mathcal{D}^{(c)}$, consisting of samples that are balanced and informative (from a meta learning perspective) and with a high prospect of being annotated with clean annotations; and 3) meta learning using $\mathcal{D}^{(t)}$ and $\mathcal{D}^{(v)}$. Each training iteration runs through these three steps.

sential contribution of INOLML is the new definition of validation set utility motivated by the bi-level objective function of meta learning algorithms. The proposed method, depicted in Fig. 3.1, consists of an iterative 3-step approach, namely: 1) pseudo-clean sample detection and robust labeling from the noisy training set; 2) validation set formation from the pseudo-clean samples from step (1), using the proposed utility criteria; and 3) meta learning using the validation set from step (2). The main contributions of the proposed method can be summarized as follows:

- A new method to build the meta learning validation set by maximizing its utility in terms of sample informativeness, class distribution balance, and label correctness;

- An innovative meta learning algorithm (Fig. 3.1), comprising these steps: 1) detection and robust labeling of pseudo-clean samples from the noisy training set; 2) formation of the validation set using the proposed utility criteria; and 3)

meta learning using the validation set from step (2).

With the technical contributions above, the evaluation of the proposed method shows significant improvements over previous meta learning approaches on imbalanced noisy-label learning benchmarks. In balanced noisy-label benchmarks, the proposed method is competitive or better than the state-of-the-art.

3.2 Background

In this section, I briefly describe a few related meta learning methods that have also attempted to solve the problem of automatically building a validation set, and their potential drawbacks. Then, I review the literature on the mixed problem of imbalanced learning and noisy-label learning.

3.2.1 Noisy-label and meta learning

In order to achieve competitive performance, the clean validation set plays a vital role in meta learning mechanisms for noisy-label learning. Traditionally, classic meta learning frameworks, such as L2W[131] and MetaWeight Net[139], find a random subset of the original training set to be annotated to form the validation set. However, not only such process can incur considerable collection and annotation costs, but they also may not select the most informative validation set for the meta learning optimization. Additionally, the generalization of the model can be damaged because such manually-curated validation is fixed for the whole training process. Despite the issues listed above, we are not aware of any previous works that inquire into the utility of validation sets for meta learning methods.

Until recently, there have been several attempts to innovate the meta learning models such that the manual formation of the meta learning validation set is no longer necessary. For instance, Zhang and Pfister [196] proposed a meta learning approach to re-weight noisy-label training samples using a validation set automatically extracted from the original noisy dataset. Inspired by DivideMix [92], their method exploits the small loss trick to identify a pseudo clean set to form the validation set, which is iteratively updated during training. However, this approach can suffer from confirmation bias, where the same samples are repeatedly selected to build the validation set. To alleviate this confirmation bias issue, the FSR model [196] also incorporates co-training that relies on two separate models, where each model's pseudo clean set is used as the validation set for the other model. Another noteworthy method that aims to remove the necessity of manually acquired validation samples is FAMUS [171], which relies on a pseudo-clean set sampled from the noisy-label training data to form the validation set. The methods above have been shown to be competitive with traditional meta learning models that require a manual validation set. Unfortunately, they only consider sample cleanliness as the sole criteria to form the validation set, which may result in a sub-optimal guidance for meta learning. Fig. 3.2 outlines the difference in framework design between our proposed INOLML and its main competitors methods, FSR and FAMUS.

3.2.2 Noisy-label and imbalanced learning

Most methods mentioned in Chapter 2 tried to address the noisy-label problem and the imbalanced-learning problem separately. In fact, most methodologies have been proposed for noisy-label learning, but they may fail when being tested on the com-

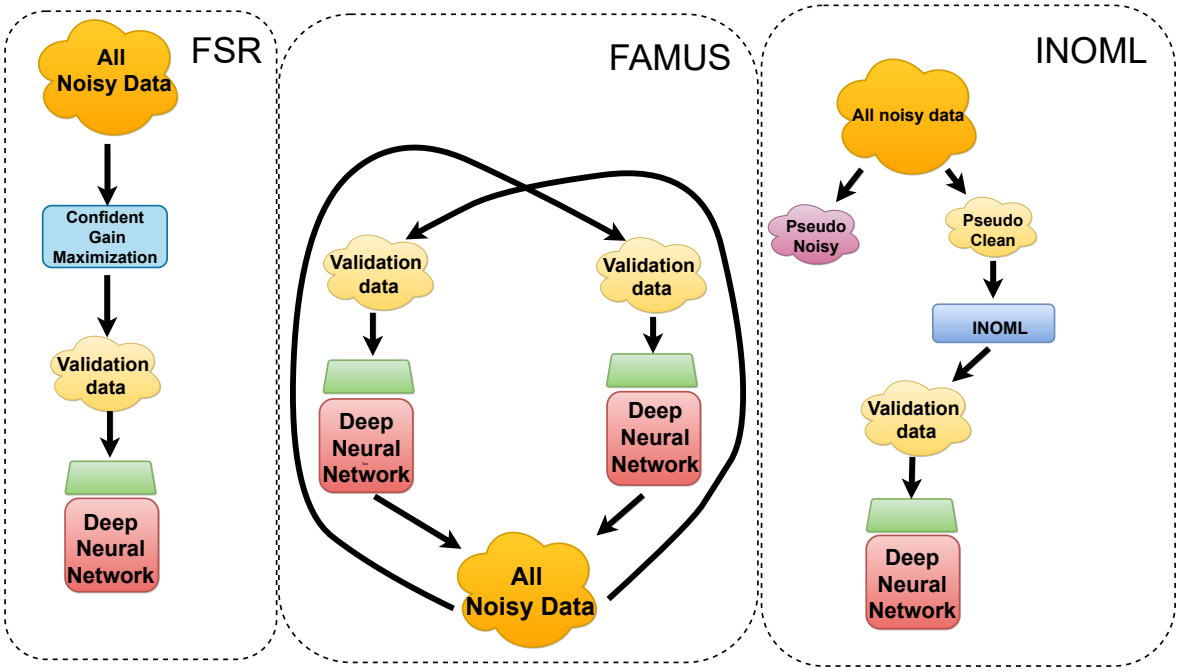


Figure 3.2: Comparison between FSR, FAMUS and our proposed INOLML.

bined problem that also contains imbalanced learning. In practice, meta learning has shown to be one of the few methods designed to handle both problems. Since meta learning models work with a balanced and clean validation set, their final target is the optimization of a model that is in principle robust to noisy labels and data imbalances. However, this theoretical design has not been empirically verified by most meta learning methods, with the exception of FSR [196]. This combined problem has been investigated by non meta learning approaches [16, 75, 162], but they fail to achieve competitive results compared to recently proposed meta learning approaches.

3.3 Methods

Initially, I define the whole noisy training set as $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{|\mathcal{D}|}$, where $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^{H \times W \times R}$ denotes an image of size $H \times W$ pixels and R colour channels, and $\mathbf{y}_i \in$

$\mathcal{Y} = \{\mathbf{v} : \mathbf{v} \in \{0,1\}^C \text{ and } \sum_{k=1}^{C} \mathbf{v}(k) = 1\}$ is the noisy one-hot encoding of label and C is the number of classes [64]. I represent our target classification model as $f_\theta : \mathcal{X} \to \Delta_{C-1}$ parameterized by $\theta \in \Theta$, with the $C-1$ probability simplex $\Delta_{C-1} = \{\mathbf{p} : \mathbf{p} \in [0,1]^C \text{ and } \sum_{k=1}^{C} \mathbf{p}(k) = 1\}$.

The proposed algorithm (INOLML) optimizes a bi-level objective function taking advantage of a meta parameter called $\omega = \{\omega_i\}_{i=1}^{|\mathcal{D}^{(t)}|}$ $(\omega_i \geq 0)$ that represents the training samples' contribution in the cross entropy loss based on their informativeness, balanced class distribution and label cleanliness. Additionally, I use another meta parameter $\lambda = \{\lambda_i\}_{i=1}^{|\mathcal{D}^{(t)}|}$ $(\lambda_i \in [0,1])$ that weights the the deep model prediction to form the pseudo label for the noisy training data, as in $\hat{\mathbf{y}}_i(\lambda_i) = \lambda_i \mathbf{y}_i + (1-\lambda_i) f_\theta(\mathbf{x}_i)$. This meta parameter has a binary value of 0 or 1, denoting whether the pseudo label is from the model's prediction or the training label. My meta learning objective is defined as:

$$\begin{aligned} \omega^*, \lambda^* &= \arg\min_{\omega,\lambda} \frac{1}{|\mathcal{D}^{(v)}|} \sum_{(\mathbf{x}_j,\mathbf{y}_j)\in\mathcal{D}^{(v)}} \ell^{(v)}\left(\mathbf{x}_j, \mathbf{y}_j; \theta^*\left(\omega,\lambda\right)\right) \\ \text{s.t.: } \theta^*(\omega,\lambda) &= \arg\min_{\theta} \frac{1}{|\mathcal{D}^{(t)}|} \sum_{(\mathbf{x}_i,\mathbf{y}_i)\in\mathcal{D}^{(t)}} \omega_i\, \ell^{(t)}\left(\mathbf{x}_i, \hat{\mathbf{y}}(\lambda_i); \theta\right), \end{aligned} \tag{3.1}$$

where $\ell^{(v)}(\mathbf{x}_j, \mathbf{y}_j; \theta^*(\omega,\lambda)) = \ell_{\text{CE}}(\mathbf{y}_j, f_{\theta^*(\omega,\lambda)}(\mathbf{x}_j))$ is the cross-entropy (CE) loss, and $\ell^{(t)}\left(\mathbf{x}_i, \hat{\mathbf{y}}(\lambda_i); \theta\right)$ is defined below in (3.13). The proposed validation set is defined as:

$$\mathcal{D}^{(v)} = \mathsf{MaxUtility}\left(\mathcal{D}^{(c)}\right), \tag{3.2}$$

where

$$\mathcal{D}^{(c)} = \mathsf{PseudoCleanDetector}\left(\mathcal{D}\right). \tag{3.3}$$

The function MaxUtility(.) in (3.2) generates the validation set $\mathcal{D}^{(v)}$ and training set $\mathcal{D}^{(t)}$ as defined in Section 3.3.1, with $\mathcal{D}^{(t)} \cap \mathcal{D}^{(v)} = \varnothing$ and $\mathcal{D}^{(t)} \cup \mathcal{D}^{(v)} = \mathcal{D}$, and $\mathcal{D}^{(v)}$ being the final validation set with informative samples that our algorithm selects for the meta learning optimization in (3.1).

Initially, the pseudo-clean set at the first training iteration is estimated from $f_\theta(.)$ trained with early-stopping. A subset of low CE loss samples [52, 92] is extracted from the noisy training set $\mathcal{D}$ by the function PseudoCleanDetector(.), taking advantage of their given noisy labels in $\mathcal{D}$ and the predictions by $f_\theta(.)$. The left out samples from $\mathcal{D}$ forms the noisy set $\mathcal{D}^{(n)}$, with $\mathcal{D}^{(c)} \cap \mathcal{D}^{(n)} = \varnothing$ and $\mathcal{D}^{(c)} \cup \mathcal{D}^{(n)} = \mathcal{D}$. This step is periodically executed throughout the training process to keep the validation set informative and meaningful. Below, we describe how to select the validation set by maximizing its utility in terms of informativeness, label cleanliness and class balance.

3.3.1 Maximizing the Utility of the Validation Set

My idea for the optimization of the utility of the validation set is inspired by the bi-level meta learning mechanism for noisy label in (3.1), which consists of the following two stages. First, I take an approximation of the optimized model parameter $\theta^*(\omega, \lambda)$ with an stochastic gradient descent (SGD) step on the training set $\mathcal{D}^{(t)}$, with each step defined by:

$$\hat{\theta}(\omega, \lambda) = \theta(\omega, \lambda) - \eta_\theta \nabla_\theta \left(\frac{1}{|\mathcal{D}^{(t)}|} \sum_{(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}^{(t)}} \omega_i \, \ell^{(t)}(\mathbf{x}_i, \hat{\mathbf{y}}_i(\lambda_i); \theta) \right) \Bigg|_{\theta = \theta(\omega, \lambda)} . \tag{3.4}$$

Subsequently, the meta parameters in the upper optimization formula, ω and λ, are estimated by a single SGD step on the validation set $\mathcal{D}^{(v)}$. For ω, we have:

$$\omega_i^* = \max\left(0, -\frac{\eta_\omega}{|\mathcal{D}^{(v)}|}\sum_{(\mathbf{x}_j,\mathbf{y}_j)\in\mathcal{D}^{(v)}}\frac{\partial}{\partial\omega_i}\ell^{(v)}(\mathbf{x}_j,\mathbf{y}_j;\theta^*(\omega,\lambda))\Big|_{\omega_i=0}\right), \tag{3.5}$$

where λ is updated according to (3.12), defined later in this chapter. The hyper-parameter ω, denoting the training sample importance, is determined by the magnitude of its gradient with respect to the cross entropy loss over the clean and informative validation set if the gradient has positive value, otherwise the framework concludes that it is a noisy or uninformative sample, hence setting the value of ω to 0. The value of ω is then normalized to have sum equal to 1. After that, I integrate the estimated meta-parameters into the lower-level optimization of (3.1) and carry out a gradient descent step to update the target model.

From [131], using a multiple layer perceptron (MPL) network, the gradient of the meta model loss w.r.t. ω is:

$$\sum_{(\mathbf{x}_j,\mathbf{y}_j)\in\mathcal{D}^{(v)}}\frac{\partial}{\partial\omega_i}\ell^{(v)}(\mathbf{x}_j,\mathbf{y}_j;\theta^*(\omega,\lambda))\Big|_{\omega_i=0} \propto -\sum_{(\mathbf{x}_j,\mathbf{y}_j)\in\mathcal{D}^{(v)}}\sum_{l=1}^{L}(\mathbf{z}_{j,l-1}^{(v)\top}\mathbf{z}_{i,l-1}^{(t)})(\mathbf{g}_{j,l}^{(v)\top}\mathbf{g}_{i,l}^{(t)}), \tag{3.6}$$

in which I define $\mathbf{z}_{j,l-1}^{(v)}$ as the deep feature output for validation image $\mathbf{x}_j$ corresponding to the $l^{(th)}$ layer of the deep model (similarly for the deep feature $\mathbf{z}_{i,l-1}^{(t)}$ of the training image $\mathbf{x}_i$). In addition, $\mathbf{g}_{j,l}^{(v)}$ denotes the gradient from the $l^{(th)}$ layer of the model for the validation sample $\mathbf{x}_j$ (the same thing applied for the gradient $\mathbf{g}_{i,l}^{(t)}$ of training image $\mathbf{x}_i$). Therefore, the training sample weight is predominantly determined by the dot product, or the similarity between its deep feature representation and gradient with other samples in the validation set. As a result, any training sample with high feature

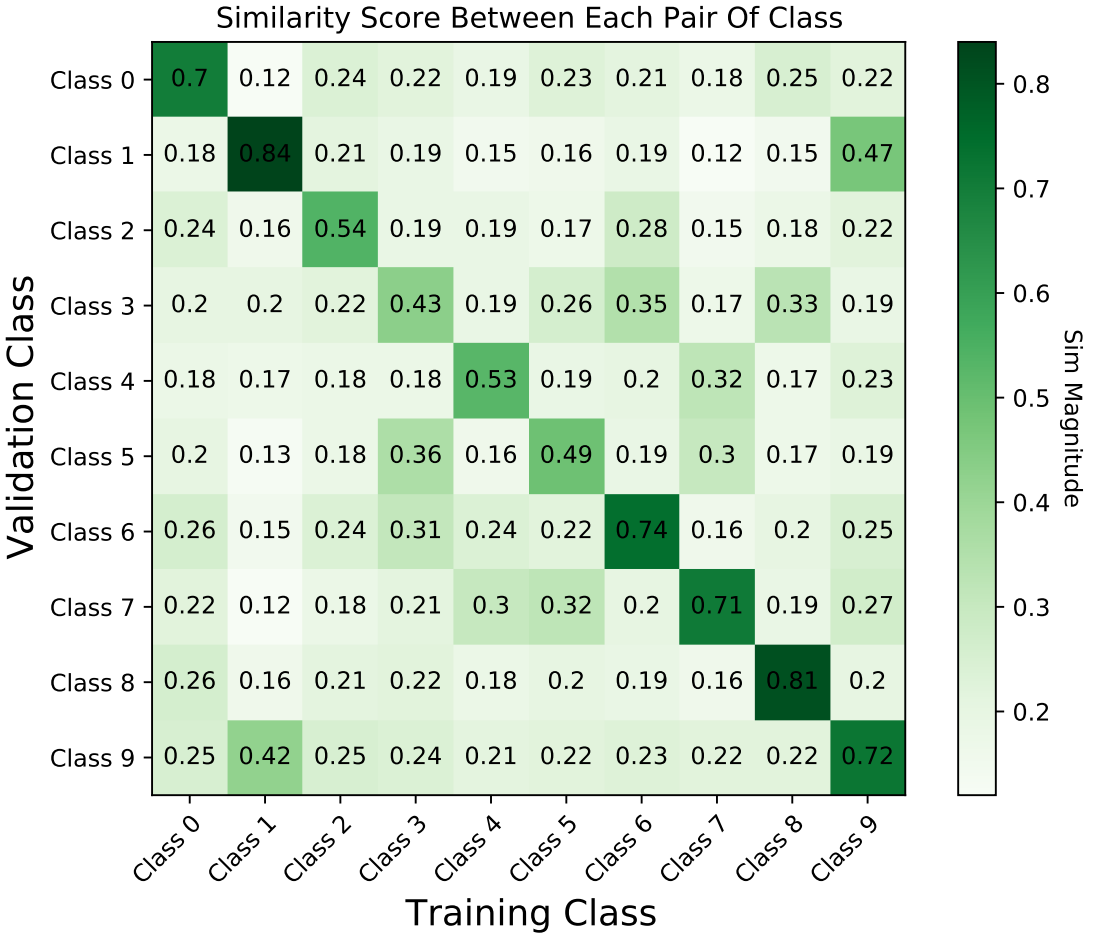


Figure 3.3: Similarity score between each pair of classes from training data from clean validation data on a CIFAR10 with 0.4 symmetric noise.

and gradient similarity with at least one of the validation samples from the same class will be rewarded for the cross entropy loss; otherwise, that sample's weight will be penalized in the final loss.

I carry out a simple experiment to analyze the correlation between validation samples with the training sample weight. Firstly, I initially trained a DivideMix model [92] on the noisy CIFAR10 dataset with 0.4 uniform noise rate. After that, I randomly selected 100 noisy-label training samples and 10 clean validation samples. Subsequently,

the DivideMix model is utilized to calculate the contribution ratio in the final weight for each class among the noisy training samples. As shown in Fig. 3.4, we can see that for each training sample, the major contribution to its weight originated from validation samples that belong to the same class. This is due to the low feature similarity of training data between different classes in the well-trained DivideMix model, making the similarity of each training sample with validation data from other classes close to zero. Fig. 3.3 proves this claim by demonstrating the corresponding of feature similarity magnitude between each couple of classes between training data and validation, in which the similarity magnitude between samples from different classes are usually insignificant compared to samples from the same class. Therefore, since meta learning with noisy-label data originated from an idea of up-weighting clean samples and down-weighting noisy sample, an optimized validation set is advantageous if it can maximize the weight of clean training samples (which also minimizes the contribution of noisy training samples since the sum of weight for each mini batch is normalized to 1). This observation is crucial to my proposal for the validation set formation.

Following this idea, I propose the following 2-stage technique to optimize the validation set: 1) construct a pseudo-clean set from the training set with candidate validation samples, and 2) form a balanced validation set from the candidate samples by identifying samples that maximize the sum in (3.6). The objective heuristic for building a validation set $\mathcal{D}^{(v)} \subset \mathcal{D}^{(c)}$ can be characterized as:

$$\begin{aligned} \mathcal{D}^{(v)} &= \arg \max_{\substack{\widehat{\mathcal{D}}^{(v)} \subset \widetilde{\mathcal{D}}^{(v)} \\ |\widehat{\mathcal{D}}^{(v)}| = M \times C}} \mathsf{Clean}\left(\widehat{\mathcal{D}}^{(v)}, \mathcal{D}^{(c)}\right) \\ \text{s.t.: } \widetilde{\mathcal{D}}^{(v)} &= \arg \max_{\substack{\bar{\mathcal{D}}^{(v)} \subset \mathcal{D}^{(c)} \\ |\bar{\mathcal{D}}^{(v)}| = K \times C}} \mathsf{Info}\left(\bar{\mathcal{D}}^{(v)}, \mathcal{D}^{(c)}\right). \end{aligned} \tag{3.7}$$

The function Info(.) in the lower-level of (3.7) is defined as:

$$\mathsf{Info}(\bar{\mathcal{D}}^{(v)}, \mathcal{D}^{(c)}) = \sum_{(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}^{(c)} \setminus \bar{\mathcal{D}}^{(v)}} \max_{\substack{(\mathbf{x}_j, \mathbf{y}_j) \in \bar{\mathcal{D}}^{(v)} \\ \mathbf{y}_j = \mathbf{y}_i}} \iota(\mathbf{x}_i, \mathbf{x}_j), \tag{3.8}$$

with

$$\iota(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l=1}^{L} (\mathbf{z}_{j,l-1}^{\top} \mathbf{z}_{i,l-1})(\mathbf{g}_{j,l}^{\top} \mathbf{g}_{i,l}), \tag{3.9}$$

in which I denote $\mathbf{z}_{j,l-1}$, $\mathbf{g}_{j,l}$ as respectively the deep feature input of $\mathbf{x}_j$ to layer l^{th} and the validation image gradient of layer l from $\mathbf{x}_j$ (same for $\mathbf{z}_{i,l-1}$ and $\mathbf{g}_{i,l}$ from $\mathbf{x}_i$). It is noteworthy that $\iota(.)$ in (3.9) from (3.6) represents the weight contributed to training sample $(\mathbf{x}_i, \mathbf{y}_i)$ by the validation sample $(\mathbf{x}_j, \mathbf{y}_j)$, or the amount of "information" that training sample $(\mathbf{x}_i, \mathbf{y}_i)$ and validation sample $(\mathbf{x}_j, \mathbf{y}_j)$ share.

Intuitively, the essence of the lower-level summation in (3.7) is the formation of a balanced validation set $\widetilde{\mathcal{D}}^{(v)}$ by maximizing the maximum shared "information" between the pseudo-clean samples from $\mathcal{D}^{(c)} \setminus \widetilde{\mathcal{D}}^{(v)}$ and any validation sample in $\widetilde{\mathcal{D}}^{(v)}$. Note that the maximum is maximized instead of the average "information" shared between training samples and validation samples in order to ensure the diversity of the validation set. The maximization of the shared "information" that each training sample can get with one of the validation samples can also guarantee that any training data get downweighted or upweighted by at least one of the clean validation samples. We illustrate this idea in Fig. 3.5.

If the whole clean validation set using the utility "information content" in (3.8) is optimized, the final validation set will contain samples with nearly duplicate information given that many neighboring samples can maximize such utility during the validation sample selection. On the other hand, by gradually identifying validation

Figure 3.4: Contribution of the training sample weight for each training class from clean validation samples for all classes on a CIFAR10 with 0.4 symmetric noise.

samples that optimize the maximization gains in each step, I can greedily select a balanced and diverse subset with little information duplication among samples. Also, it is intuitive that the validation set optimization leads to an improvement in the prediction quality of the clean training samples that have high similarity with validation samples since deep neural network tends to produce similar results for similar input samples, and similar training samples are more likely to have similar ground-truth labels.

Note that while the informativeness of the validation set is important, it is also

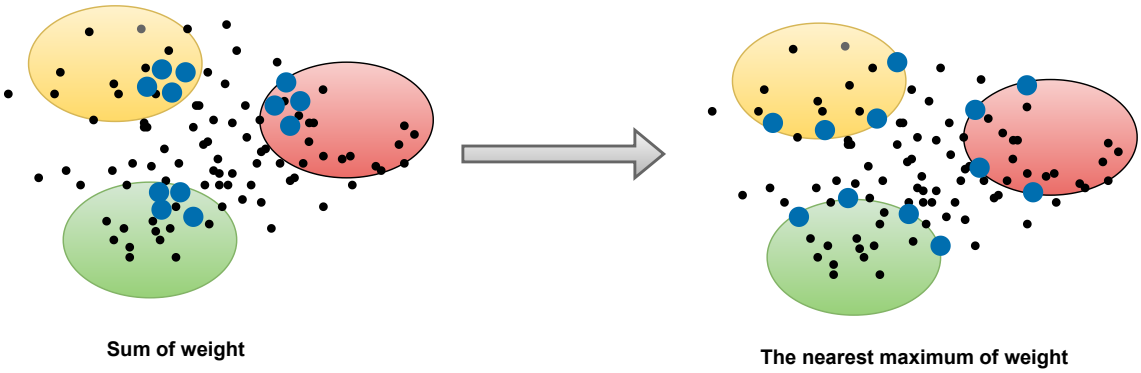


Figure 3.5: Difference between the optimization of the average (or sum) of "information content" (left figure) and the maximization of the maximum "information content" (figure on the right) using a 3-class classification problem, with the likelihood of each class represented by ellipsoids of different colors. Selecting a validation sample that maximizes ω in Eq. (3.6) leads to a validation set containing nearly duplicate data (figure on the left), while selecting them one by one by optimizing Eq. (3.8) generates a more diverse validation set (right hand side).

necessary to ensure the cleanliness of samples. The samples in $\widetilde{\mathcal{D}}^{(v)}$ are not completely clean since $\mathcal{D}^{(c)}$ is not guaranteed to be noise-free and at the same time, the function $\mathsf{Info}(.)$ can reward samples in $\widetilde{\mathcal{D}}^{(v)}$ that have low logit scores and large gradients. Therefore, despite prioritizing informativeness samples over clean ones, reducing label noise rate in the validation set is also valuable. An intuitive observation from our analysis indicates that the probability of samples from $\widetilde{\mathcal{D}}^{(v)}$ to be clean is proportional to the similarity magnitude they have with other samples of the same class. Hence, originated from the similarity between the sample of interest and other samples of the same class in $\mathcal{D}^{(c)}$, I define an heuristic to mitigate the noise ratio of the selected validation , defined as follows:

$$\mathsf{Clean}\left(\widehat{\mathcal{D}}^{(v)}, \mathcal{D}^{(c)}\right) = \sum_{\substack{(\mathbf{x}_j,\mathbf{y}_j)\in\widehat{\mathcal{D}}^{(v)} \\ (\mathbf{x}_i,\mathbf{y}_i)\in\mathcal{D}^{(c)}\setminus\widehat{\mathcal{D}}^{(v)} \\ \mathbf{y}_i=\mathbf{y}_j}} \sum_{l=1}^{L} \left(\mathbf{z}_{j,l-1}^{\top}\mathbf{z}_{i,l-1}\right). \tag{3.10}$$

To guarantee a balanced class distribution for $\mathcal{D}^{(v)}$, I add a constraint that selects M samples for each of the C classes with $M \ll K$, as shown in the upper-level of (3.7).

Both the combinatorial optimizations in (3.7) are solved with greedy strategies, where I loop through each class and select K samples per class that maximize the lower objective function, followed by sequentially selecting M samples among the previous set of K samples for each class by optimizing the upper objective function. In practice, I initially limit the potential candidates to only the upper half of the samples that maximize the sum of the utility in (3.8), before gradually identifying samples that maximize the upper objective function of (3.7). If there exists multiple candidates that either give us the same value for the upper objective function when adding them to our validation set, or none of the available samples can boost the value of our upper objective function, then among those candidate samples, we select the ones that maximize the lower objective function formula. Fig. 3.6 summarizes step-by-step the proposed methodology to find the clean validation set.

Also, I simplify the calculation of gradient in (3.6) and the optimization in (3.7) by using only on the last layer L of the model because, according to [196], the weights of training samples in meta learning depend mostly on the last layer of the model. Fig. 3.7 shows the magnitude of gradient by some of the last layers of a deep neural network model. Based on the graph, the gradient magnitude of the last layers tend to be larger than for other layers. This is due to the chain rule of gradient descent that reduces the gradient magnitude. Hence, instead of using the formula in (3.9) for weight estimation, I use a simplified version that uses only the contribution from the final fully connected layer, as in:

$$\iota(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{z}_{j,L-1}^\top \mathbf{z}_{i,L-1})(\mathbf{g}_{j,L}^\top \mathbf{g}_{i,L}). \tag{3.11}$$

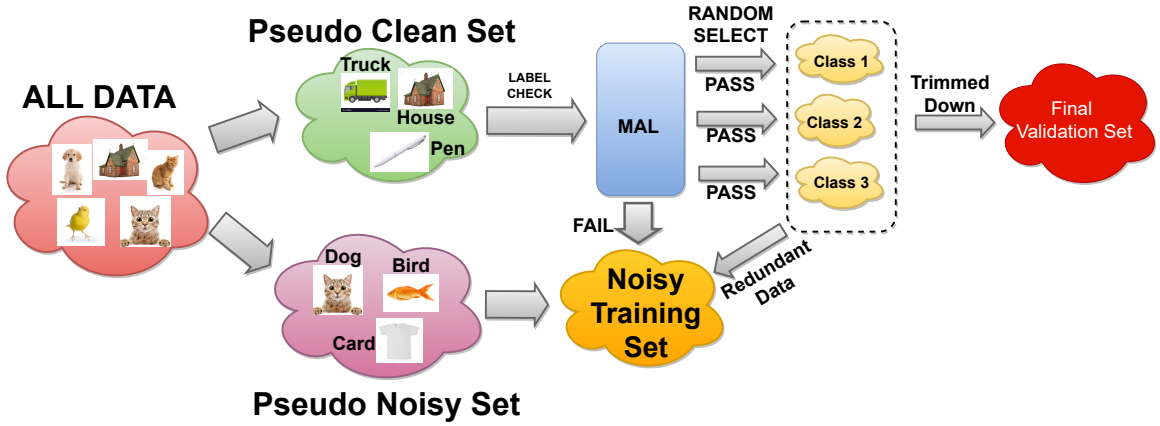


Figure 3.6: Summary of the process of building the validation set. MAL denotes our moving average label consistence test, in which training samples that fail the test will be included in the noisy training set while among training samples that passed the inspection, 200 of them will be randomly sampled to be candidates for the new validation set optimization. The training samples that are not selected for the validation set will be grouped with the noisy training set.

Another reason that explains the motivation of using only the last layer in the computation of (3.11) is the large memory complexity involved in using all network layers. For instance, around $K = 200$ candidate samples per class are typically needed for $\tilde{\mathcal{D}}^{(v)}$, which is later trimmed to $M = 10$ to form $\mathcal{D}^{(v)}$ (e.g., for CIFAR100, 200 samples per class are required, so for 100 classes we will need 20000 samples for the candidate set). For each of these samples, we need to infer their deep feature representation and gradient from the deep neural network, at every layer, for our validation set optimization. Consequently, even if we use a small Resnet18 with around 11×10^6 parameters, a massive amount of space of up to $(2 \times 10^4) \times (11 \times 10^6)$ floating point numbers is necessary for storing this information, which is very expensive and generally intractable.

Fig. 3.8 visualizes changes in the validation set during training using a toy example built from clusters generated by three different Gaussian models representing three

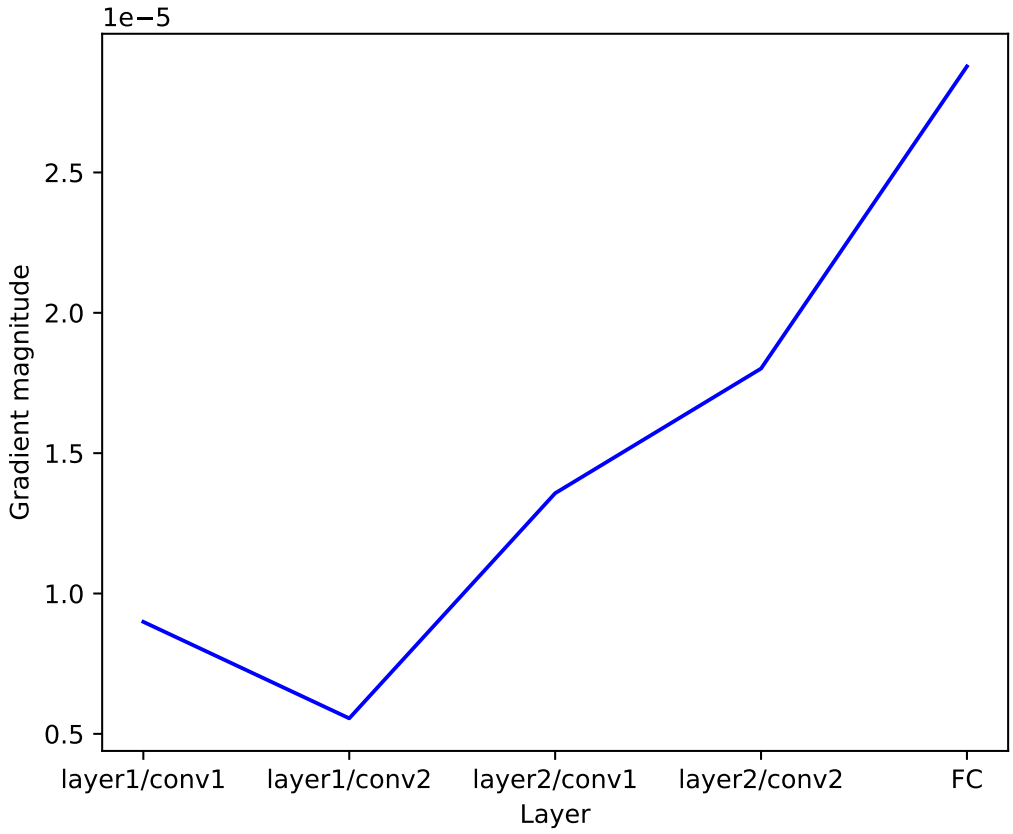


Figure 3.7: Magnitude of gradients from some of the last layers of a deep neural network.

classes in the feature space. In this experiment, samples with large gradient magnitude are selected, so informative samples will have high probability to be selected. At the beginning of the training, a randomly initialized model can only generate random outputs for all training inputs. Thus, "easy" samples (i.e., data that are easily identified by the models at the center of the clusters) will be selected for the validation set since they have high similarity with samples of the same class. As the training progresses, the model gets better predictions, where informative samples will gradually be represented by low confidence samples located at the boundary between classes. Intuitively, the

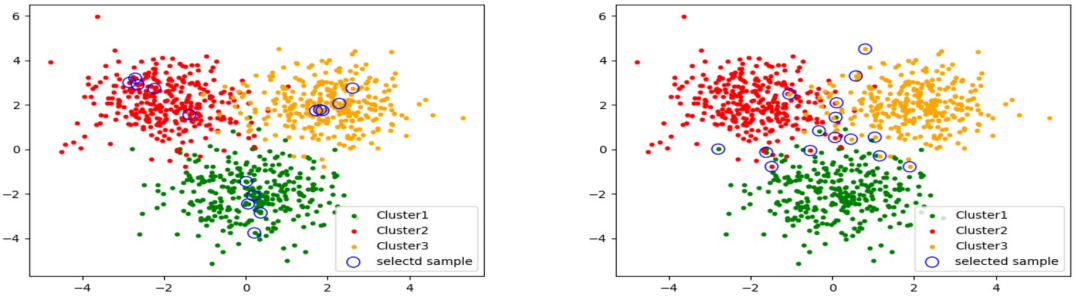


Figure 3.8: Selected samples to be included in the validation set at the beginning (left graph) and the end (right graph) of the training.

validation samples at later training stages can identify the boundaries between classes.

3.3.2 Training Procedure

We depict our 3-step methodology in Fig. 3.1. The methods starts with the pseudo-clean label detector (step 1), followed by the maximization of the validation set utility (step 2), and finishes with the meta learning optimization based on (3.1) with the loss $\ell(.)$ defined in [197].

Similar to ω, the second meta parameter λ^* in (3.1) is estimated with a single gradient descend, except that λ^* exploits the gradient directly with [197]:

$$\lambda_i^* = \left[\text{sign} \left(\sum_{(\mathbf{x}_j, \mathbf{y}_j) \in \mathcal{D}^{(v)}} \frac{\partial}{\partial \lambda_i} \ell^{(v)}(\mathbf{x}_j, \mathbf{y}_j; \theta^*(\omega, \lambda)) \right) \right]_+ , \tag{3.12}$$

with $\ell^{(v)}(\mathbf{x}_j, \mathbf{y}_j; \theta^*(\omega, \lambda))$ being previously defined in (3.1). Once we finish the estimation of ω^* and λ^*, the lower formula of (3.1) is optimized with SGD to estimate the

model parameter with the following loss function [197]:

$$\ell^{(t)}(\mathbf{x}_i, \mathbf{y}_i; \theta) = \omega_i^* \ell_{\mathrm{CE}}(\widehat{\mathbf{y}}_i(\lambda_0), f_\theta(\mathbf{x}_i)) + \frac{1}{B}\ell_{\mathrm{CE}}(\mathbf{y}_i^*(\lambda_i^*), f_\theta(\mathbf{x}_i)) + p \times \ell_{\mathrm{CE}}(\mathbf{y}_i^\beta, f_\theta(\mathbf{x}_i^\beta)) + k \times \ell_{\mathrm{KL}}(f_\theta(\mathbf{x}_i), f_\theta(\mathsf{Augment}(\mathbf{x}_i))), \tag{3.13}$$

where $\widehat{\mathbf{y}}_i(\lambda_0)$ represents the pseudo-label incorporated with λ_0 from (3.1). The value of hyperparameter λ_0 is kept fixed with a value of 0.9, while p and k are hyper-parameters for the contribution of the MixUp loss utilizing unlabeled and labeled samples, while B is the batch size. Additionally, the final pseudo label $\mathbf{y}_i^*(\lambda_i^*)$ for cross-entropy training is defined as $\mathbf{y}_i^*(\lambda_i^*) = \mathbf{y}_i$, if $\lambda_i^* > 0$; otherwise $\mathbf{y}_i^*(\lambda_i^*) = f_\theta(\mathbf{x}_i)$, $\mathbf{y}_i^\beta$ and $\mathbf{x}_i^\beta$ are respectively the MixUp labels and images obtained when we employ a mixup linear combination operator [187] between samples in the training, their augmentation set $\mathcal{D}_a^{(t)}$ and the validation set as follows:

$$\mathbf{x}_i^\beta = \mathsf{Mix}_\beta(\mathbf{x}_a, \mathbf{x}_b), \tag{3.14}$$

$$\mathbf{y}_i^\beta = \mathsf{Mix}_\beta(\mathbf{y}_a, \mathbf{y}_b), \tag{3.15}$$

$$\text{with } (\mathbf{x}_a, \mathbf{y}_a), (\mathbf{x}_b, \mathbf{y}_b) \in \mathcal{D}^{(t)} \cup \mathcal{D}_a^{(t)} \cup \mathcal{D}^{(v)}, \tag{3.16}$$

We also adopted the Kullback-Leibler (KL) divergence [87] in (3.13) for the regularization between the model output for training image $\mathbf{x}_i$ and its data augmentation $\mathsf{Augment}(\mathbf{x}_i)$.

The quality of the meta learning optimization depends on the actual (hidden) proportion of clean samples in the pseudo clean set $\mathcal{D}^{(c)}$, while the efficiency depends on the size of $\mathcal{D}^{(c)}$. Hence, to reduce computational cost, the selection of the validation set in (3.7) uses the subset $\tilde{\mathcal{D}}^{(c)} = \{(\mathbf{x}_i, \mathbf{y}_i) : (\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}^{(c)} \wedge \arg\max_{k \in \{1,\ldots,C\}} \mathbf{y}_i(k) =$

$\arg\max_{k\in\{1,\dots,C\}} \tilde{\mathbf{y}}_i(k)\}$ instead of the whole pseudo clean set $\mathcal{D}^{(c)}$. This subset comprises N randomly-sampled images $(\mathbf{x}_i, \mathbf{y}_i)$ for each class belonging to $\mathcal{D}^{(c)}$ that can maintain the consistency between their noisy annotation $\mathbf{y}_i$ and the corresponding moving average robust label, defined as the average model response over the last E epochs:

$$\tilde{\mathbf{y}}_i = \kappa\tilde{\mathbf{y}}_i + (1-\kappa)\frac{1}{E}\sum_{e=1}^{E} f_\theta(\mathbf{x}_i), \tag{3.17}$$

where $\kappa \in [0, 1]$ is a hyper-parameter. So for example, using CIFAR10, which has 50K training samples, with 40% symmetric noise, our validation set optimization will be expensive if the potential candidates can come from the whole $D^{(c)}$ since the inference from the model will be needed for around 30K samples.Instead of that, if we set $N = 200$, a random subset of size $200 \times C$ with C being the number of classes, will be extracted from $D^{(c)}$ as a simplified representation for $D^{(c)}$. Since CIFAR10 has 10 classes, we are only required to obtain inferences for around 2000 training samples, which is less then 10% of the 30K samples above. Additionally, we only consider the samples in $D^{(c)}$ that have consistent noisy labels and moving average labels. The details of the training process are in Algorithm 2.

3.4 Results

3.4.1 Datasets

For the experiments, we utilize common datasets for noisy label benchmarks, including both synthetic and real-world label noise, including: CIFAR10, CIFAR100, CNWL and Webvision. CIFAR10 and CIFAR100 [86] are medium-size datasets with 50K images for training and 10K images for testing. The training images for both datasets are of

Algorithm 2 Training procedure of the proposed INOLML.

procedure TRAINING($\mathcal{D}$, η, T, $\widetilde{T}$, $T^{(u)}$, $\widetilde{\eta}$, κ, N, M, K, C)
  ▷ *$\mathcal{D}$: noisy training set, $\{\eta_t\}_t^T$: learning rates* ◁
  ▷ *T: total number of iterations* ◁
  ▷ *$\widetilde{T}$: minimum number of iterations before updating the moving average robust labels* ◁
  ▷ *$T^{(u)}$: interval between updates* ◁
  ▷ *$\widetilde{\eta}$: learning rate threshold* ◁
  ▷ *κ, N, M, K, C: hyper-parameters* ◁
  Warmup $f_\theta(.)$ with $\ell_{\text{CE}}(.)$ from $\mathcal{D}$
  $\mathcal{D}^{(c)} = \mathsf{PseudoCleanDetector}\,(\mathcal{D})$ using (3.3)
  Initialise moving average robust label $\{\tilde{\mathbf{y}}_i\}_{i=1}^{|\mathcal{D}^{(c)}|}$ of samples in $\mathcal{D}^{(c)}$
  Initialise $\mathcal{D}^{(v)}$ and $\mathcal{D}^{(t)}$ from $\mathcal{D}^{(c)}$ using (3.7)
  Reinitialize $f_\theta(.)$
  for $t = 1$ to T **do**
    Meta-learn to train θ, ω and λ using (3.1)
    if $(t > \widetilde{T})$ and $(\eta_t < \widetilde{\eta})$ **then**
      Update $\{\tilde{\mathbf{y}}_i\}_{i=1}^{|\mathcal{D}^{(c)}|}$ of samples in $\mathcal{D}^{(c)}$
    if $\eta_t = 0$ **then**
      $\mathcal{D}^{(c)} = \mathsf{PseudoCleanDetector}\,(\mathcal{D})$ from (3.3)
    if $t \mod T^{(u)} = 0$ **then**
      Update $\mathcal{D}^{(v)}$ and $\mathcal{D}^{(t)}$ from $\mathcal{D}^{(c)}$ (3.7)
  return the trained model parameter θ

size 32×32 pixels, and have labels belonging to 10 or 100 classes, respectively.

Webvision [94] is a large scale dataset of 2.4 million images obtained from crawling Google and Flickr, where each image has size 224×224 pixels and is annotated with a noisy label belonging to one of the 1,000 classes from ImageNet [33]. This dataset is more challenging than CIFAR, not only due to its larger size and number of classes, but also because of class imbalance and real-world noisy labels. This benchmark uses a subset of Webvision containing the first 50 classes to construct the new Webvision mini dataset. The CNWL [71] dataset has around 50K training images and 5K testing images belonging to ones of 100 classes.

The noise rates of the datasets above range from 0 to 0.8.

3.4.2 Implementation Details

The hyperparameters for our method as well as the network architecture are the same as the base meta learning framework Distill [197]. Similarly to Distill [197], we also use of the cosine learning rate decay with warm restarting [107] and SGD optimizer. For CIFAR datasets with synthetic noise, we use two different architecture as the base model: the big model WideResnet28-10 and a smaller model Resnet29, for fair comparison with [197]. However, with CIFAR synthetic imbalanced setting scenarios, we use the Resnet32 architecture in order to compare with FAMUS [171]. For the real-world dataset Webvision, we follow FSR [196] and run a single Resnet50 model as the base framework for Webvision, while for CNWL, we adopted the PreAct Resnet18, following previous works [26, 122] on this benchmark. The batch size for all benchmarks, except Webvision, is 100, while we use a batch size of 16 for Webvision due to the bigger volume of input images. In terms of the number of iterations, we use 1 millions iterations, 100 thousand iterations, and 150 thousand iterations for Webvision, CIFAR10/CIFAR100 and CNWL experiments .

The experiment results are reported in terms of the prediction accuracy of on their corresponding testing sets. We compare our method with several existing state-of-the-art (SOTA) meta learning methods, including FaMUS [171], FSR [196], Meta Weight Net [139], Distill [197], GDW [18], L2R[131]. Additionally, we also provide the result of SOTA non meta learning noisy-label learning approaches, such as Divide Mix [92], CausalNL [175], NPC [8], MentorMix [71], MentorNet [70], MOIT [122], GJS [37], CRUST [115], Co-teaching [52], Iterative-CV [19], HAR [16], D2L [108], PENCIL [176]

Table 3.1: Test accuracy (%) of our INOLML and previous methods for symmetric noise. Methods with ^{T} represent meta learning methods that need clean validation sets. The lower block contains meta learning methods, while the upper block shows SOTA methods.

Method	CIFAR10			CIFAR100		
	0.2	**0.4**	**0.8**	**0.2**	**0.4**	**0.8**
GJS	95.3 ± 0.2	93.6 ± 0.2	79.1 ± 0.3	78.1 ± 0.3	75.7 ± 0.3	44.5 ± 0.5
DivideMix	95.7 ± 0.0	-	92.9 ± 0.0	76.9 ± 0.0	-	59.6 ± 0.0
CRUST	91.1 ± 0.2	89.2 ± 0.2	58.3 ± 1.8	-	-	-
PENCIL	-	-	-	73.9 ± 0.3	69.1 ± 0.6	-
ELR	92.1 ± 0.4	91.4 ± 0.2	80.7 ± 0.6	74.7 ± 0.3	68.4 ± 0.4	30.2 ± 0.8
CausalNL + NPC	81.2 ± 0.0	-	18.8 ± 0.0	-	-	-
Distill^{T}	96.2 ± 0.2	95.9 ± 0.2	93.7 ± 0.5	81.2 ± 0.7	80.2 ± 0.3	**75.5 ± 0.2**
MentorNet^{T}	92.0 ± 0.0	89.0 ± 0.0	49.0 ± 0.0	73.0 ± 0.0	68.0 ± 0.0	35.0 ± 0.0
L2R^{T}	90.0 ± 0.4	86.9 ± 0.2	73.0 ± 0.8	67.1 ± 0.1	61.3 ± 2.0	35.1 ± 1.2
MWN^{T}	90.3 ± 0.6	87.5 ± 0.2	-	64.2 ± 0.3	58.6 ± 0.5	-
GDW^{T}	-	88.1 ± 0.4	-	-	59.8 ±1.6	-
FaMUS	-	95.3 ± 0.2	-	-	76.0 ± 0.2	-
FSR	95.1 ± 0.1	93.7 ± 0.1	82.8 ± 0.3	78.7 ± 0.2	74.2 ± 0.4	46.7 ± 0.8
INOLML	**96.9 ± 0.1**	**96.6 ± 0.1**	**95.0 ± 0.2**	**82.0 ± 0.2**	**81.3 ± 0.2**	74.7±0.1

and ELR [102]. For imbalanced data contaminated with noise experiments, we include some recent prominent noisy-label imbalanced learning methods to compare with, including ROLT [162], FSR [196], LDAM [15], BBN [203], HAR [16], and CRUST [115].

3.4.3 Symmetric Noise

We display the performance of our model and other methods, including meta learning approaches and recent non meta learning SOTA methods, on CIFAR10 and CIFAR100 datasets with synthetic symmetric noise benchmarks in Table 3.1. Following other methods, we use noise rates from 20% to 80%. Previous approaches that require a clean validation set is indicated with ^{T}.

Overall, our proposed INOLML demonstrates significant boost in performance com-

pared to all previous methods, in the majority of benchmarks. The only exception is CIFAR100 with 80% noise, where Distill shows better results mostly because it has access to a large clean validation set, containing 10 images per class. In large noise rate problems, such as this one with 80% noise, our validation set gets severely contaminated, making the model vulnerable to the accumulated noisy-labels in the validation set. Fig. 3.9a shows the issue that for 80% symmetric noise ratio, the noisy-label samples consist of a considerable part (20% to 45%) of the validation set, which results in the deterioration of our approach's efficacy. Additional experiments with various validation set sizes ranging from 1 to 10 samples per class is carried out and shown in Table 3.2. Based on the results shown in this table, our method has achieved significant improvement, outperforming the Distill model with by 1% to 3% in most scenarios.

In summary, all symmetric noise experiments show empirical evidence that our proposed utility to build the validation set can be beneficial for the meta learning process. Additionally, the results indicate the supremacy of our method even under high noise rate scenarios, despite the presence of noise in the validation set. Moreover, our proposed heuristic has set new SOTA results on most symmetric noise benchmarks for CIFAR10 and CIFAR100 datasets, compared to methods that do not need clean validation set, outperforming them by remarkable margins.

3.4.4 Asymmetric Noise

Similar to symmetric noise, we also compared our method's performance for asymmetric noise benchmarks with other existing meta learning and non meta learning SOTA models. However,we only use a single noise ratio of 0.4, in which the result is displayed in Tables 3.3 and 3.4. From Table 3.3, we can see the comparison in test accuracy

Table 3.2: Test accuracy (in %) comparison between our method (INOLML) and the Distill model (DN) on symmetric noise (rates of 20%, 40% and 80%) using 1, 5 and 10 samples per class in the validation set on two backbone models: Resnet29 (RN29) and Wideresnet28-10 (WRN). The results of the Distill model with WideResnet28-10 are collected from [197]. Recall that the Distill needs a clean validation set, while our INOLML works with an automatically built validation set.

Method	**Val. Set size**	**Dataset**					
		CIFAR10			**CIFAR100**		
		0.2	**0.4**	**0.8**	**0.2**	**0.4**	**0.8**
DN-RN29	1	87.0 ± 0.5	85.3 ± 0.5	FAIL	58.9 ± 0.5	55.8 ± 0.7	FAIL
INOLML-RN29		90.3 ± 0.2	89.1 ± 0.5	79.1 ± 0.3	65.9 ± 0.2	61.5 ± 0.2	55.1 ± 0.1
DN-RN29	5	90.7 ± 0.3	89.0 ± 0.3	83.5 ± 0.2	62.6 ± 0.4	58.8 ± 0.5	48.5 ± 0.5
INOLML-RN29		90.9 ± 0.2	90.9 ± 0.1	87.4 ± 0.2	66.6 ± 0.1	65.7 ± 0.1	59.0 ± 0.5
DN-RN29	10	91.0 ± 0.2	89.2 ± 0.1	87.0 ± 0.1	63.7 ± 0.2	60.5 ± 0.2	57.5 ± 0.5
INOLML-RN29		92.2 ± 0.1	91.0 ± 0.1	87.9 ± 0.2	67.1 ± 0.1	66.3 ± 0.1	59.2 ± 0.2
DN-WRN	1	95.4 ± 0.6	94.5 ± 1.0	87.9 ± 5.1	77.4 ± 0.4	75.5 ± 1.1	62.1 ± 1.2
INOLML-WRN		96.0 ± 0.2	95.9 ± 0.2	94.3 ± 0.2	81.6 ± 0.2	79.5 ± 0.2	73.6 ± 0.3
DN-WRN	5	96.4 ± 0.0	95.5 ± 0.6	91.8 ± 3.0	80.4 ± 0.5	79.6 ± 0.3	73.6 ± 1.5
INOLML-WRN		96.4 ± 0.1	96.2 ± 0.1	94.6 ± 0.2	82.2 ± 0.2	81.5 ± 0.2	74.5 ± 0.2
DN-WRN	10	96.2 ± 0.2	95.9 ± 0.2	93.7 ± 0.5	81.2 ± 0.7	80.2 ± 0.3	**75.5 ± 0.2**
INOLML-WRN		**96.9 ± 0.1**	**96.6 ± 0.1**	**95.0 ± 0.2**	**82.0 ± 0.2**	**81.3 ± 0.2**	74.7 ± 0.1

between our INOLML and Distill given various validation set sizes of 1, 5 or 10 sample per class, while the results in Table 3.4 show our model's performance compared to other non meta learning SOTA models. Even though INOLML does not need clean validation data, the model still attains promising performance, which is significantly better than Distill's, especially for low capacity models such as Resnet29 or when the validation set is small (e.g., 1 sample per class). Unfortunately, similarly to the symmetric noise benchmark, INOLML produces slightly lower test accuracy than Distill when it has large validation sets (e.g., 5 or more sample per class) and architectures with larger capacity. This decrease in performance is due to the noise present in our pseudo-clean validation set. Unlike the random noise setting above, asymmetric noise

Table 3.3: Test accuracy (%) of our INOLML and previous methods on CIFAR10 with 0.4 asymmetric noise. Comparison with Distill using a validation set $\mathcal{D}^{(v)}$ of sizes 1, 5 and 10 samples per class on Resnet29 and WideResnet28-10. The superscript $^{\mathrm{T}}$ indicates the need for clean validation sets.

Method	$\left\|\mathcal{D}^{(v)}\right\|$	**Resnet29**	**WRN28-10**
Distill$^{\mathrm{T}}$	$1 \times C$	76.8 ± 2.9	93.2 ± 0.2
INOLML		86.8 ± 0.9	93.6 ± 0.3
Distill$^{\mathrm{T}}$	$5 \times C$	86.7 ± 0.5	94.5 ± 0.2
INOLML		89.3 ± 0.2	94.1 ± 0.1
Distill$^{\mathrm{T}}$	$10 \times C$	88.6 ± 0.7	$\mathbf{94.9 \pm 0.1}$
INOLML		$\mathbf{89.8 \pm 0.3}$	94.2 ± 0.1

Table 3.4: Test accuracy (%) of our INOLML and previous methods on CIFAR10 with 0.4 asymmetric noise. The superscript $^{\mathrm{T}}$ indicates the need for clean validation sets.

Method	**Accuracy**
GJS	89.7 ± 0.4
F-Correction	83.6 ± 0.3
PENCIL	91.2 ± 0.0
DivideMix	92.1 ± 0.0
MLNT	92.3 ± 0.1
CausalNL	74.8 ± 0.0
L2R$^{\mathrm{T}}$	90.8 ± 0.3
FSR	93.6 ± 0.3
INOLML	$\mathbf{94.2 \pm 0.1}$

only flips labels between a few classes that are highly correlated, making the model more prone to overfitting. Moreover, Table 3.3 also shows that larger architectures (e.g., WideResnet28) are more likely to fit label noisy patterns more easily due to their higher memorization capacity. Overall, our INOLML shows better performance compared to SOTA methods, such as MLNT, FSR and DivideMix in Table 3.4.

Table 3.5: Test accuracy (%) of our INOLML and previous SOTA methods for instance-dependent noise.

Method	CIFAR10			CIFAR100		
	0.2	**0.3**	**0.4**	**0.2**	**0.3**	**0.4**
Decoupling	78.71	75.17	61.73	36.53	30.93	27.85
kMEIDTM	92.26	90.73	85.94	69.16	66.76	63.46
Co-teaching	80.96	78.56	73.41	37.96	33.43	28.04
DivideMix	94.80	94.60	94.53	77.07	76.33	70.80
CausalNL	81.79	80.75	77.98	-	-	-
MentorNet	81.03	77.22	71.83	38.91	34.23	31.89
HOC	90.03	-	85.49	68.82	-	62.29
CAL	92.01	-	84.96	69.11	-	63.17
T-Revision	76.15	70.36	64.09	37.24	36.54	27.23
Reweight	76.23	70.12	62.58	36.73	31.91	28.39
PTD-R-V	76.58	72.77	59.50	65.33	64.56	59.73
INOLML	**96.53**	**96.46**	**96.40**	**81.62**	**81.09**	**80.51**

3.4.5 Instance-dependent Noise

Another important benchmark recently is the instance-dependent noise that we have defined in Chapter 2. This category of noise theoretically is more realistic and challenging. Following recent state-of-the-art methods in this benchmark recent, we generate the instance-dependent noise according to Algorithm 1 on CIFAR10 and CIFAR100 datasets, provide and compare the result between our methods and other SOTA approaches in meta learning, as well as recent non meta learning SOTA methods. For fair with other methods, we use a noise rate from 0.2 to 0.4 and record the result as the average performance of 3 independent runs for each benchmark. The result is demonstrated in Table 3.5.

In summary, INOLML displays considerable advancement in performance compared to all previous methods across the majority of benchmarks. Our methods surpassed the results of both recent meta and non-meta learning SOTA frameworks by a significant

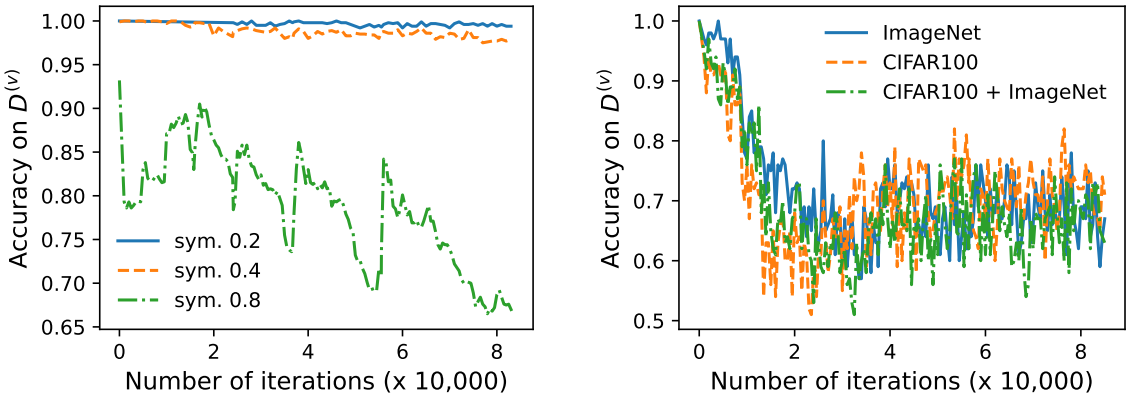
margin.

Additionally, similar to uniform noise, we run further experiments considering different validation set sizes ranging from 1 to 10 samples per class. The results of this experiments utilize the small base network Resnet29 and is demonstrated in Table 3.6. From the performance displayed in Table 3.6, despite showing slightly improvement when the validation set size is large, our INOLML is significantly more consistent in performance, even under extremely small validation set size (1 sample per class). The gap in performance between our methods and the Distill Noise framework is increased under higher noise ratio as well as smaller validation set, with a difference of 0.5% to 3% in most scenarios.

Table 3.6: Test accuracy (in %) comparison between our method (INOLML) and the Distill model (DN) on instance-dependent noise (rates of 20% and 40%) using 1, 5 and 10 samples per class in the validation set on Resnet29 model as the backbone framework (RN29). We note that Distill needs a clean validation set, while our INOLML demands the manual acquisition of an extra clean validation set.

Method	**Val. Set size**	**Dataset**			
		CIFAR10		**CIFAR100**	
		0.2	**0.4**	**0.2**	**0.4**
DN-RN29	1	86.6 ± 1.0	85.0 ± 1.0	58.9 ± 0.5	55.8 ± 0.7
INOLML-RN29		89.5 ± 0.3	87.2 ± 0.6	61.3 ± 1.0	59.9 ± 0.8
DN-RN29	5	90.2 ± 0.1	88.2 ± 0.2	62.3± 0.3	59.9 ± 0.5
INOLML-RN29		91.3 ± 0.1	90.3 ± 0.2	64.9 ± 1.0	63.4 ± 0.5
DN-RN29	10	91.0 ± 0.1	90.0 ± 0.2	63.1± 2.0	62.5 ± 0.5
INOLML-RN29		91.6 ± 0.1	90.5 ± 0.3	65.5 ± 1.0	63.7 ± 0.5

Overall, all empirical evidences point out that meta learning frameworks benefit greatly from our utility criteria to build the validation set for addressing instance-dependence noise problems. Our results are better than the current state-of-the-art by



(a) Symmetric noise benchmarks (b) Open-set noise benchmarks

Figure 3.9: Accuracy of the clean validation set $\mathcal{D}^{(v)}$ as training progresses evaluated on different noise benchmarks.

a margin of at least 2% on every instance-dependent noise benchmark. Moreover, unlike our initial prediction that the meta learning frameworks may suffer from confirmation bias with noise in the validation set, our model shows consistency in performance, with a slightly decrease of 0.5%-1% when we increase the noise rate. Please note that we still hold the advantage of memory complexity and model practicality thanks to oour single model design in comparison to popular SOTA co-training approaches such as DivideMix[92] or Co-teaching [51] that require two models.

3.4.6 Semantic Noise

One of the hardest synthetic noise type that has been used as benchmark by previous method is semantic noise. This family of noise takes advantage of a weak deep neural network to generate noisy samples, by utilizing the network's predictions as the new labeller for all training samples after warmed up the network with a subset of the original training data, according to Lee et al. [89]. These data samples with new labels

are subsequently exploited for training on another target model. As the new labels depict the class that the network are prone to fit, wrong annotations generated by such a way can trick the deep neural network, making the model less robust to label noise. In order to investigate our method's performance in combating this type of noise, we generate semantic noise on CIFAR10 and CIFAR100, with a noise ratio of 31% and 37% respectively. We also run experiments with validation sets of various sizes (1, 5 and 10 samples per class), and compare our method with the Distill baseline model. Table 3.7 displays the result of our method in comparison with Distill.

Table 3.7: Test accuracy (in %) comparison between our method (INOLML) and the Distill model (DN) on semantic noise (rates of 31% for CIFAR10 and 37% for CIFAR100) using 1, 5 and 10 samples per class in the validation set on Resnet29 (RN29) and WideResnet28-10(WRN) model as the backbone frameworks. The results are recorded by taking the average performance of 3 independent runs.

Method	**Val. Set size**	**Dataset**	
		CIFAR10	**CIFAR100**
		0.31	**0.37**
DN-RN29	1	82.1	66.6
INOLML-RN29		85.3	87.6
DN-RN29	5	88.0	73.0
INOLML-RN29		87.0	73.0
DN-RN29	10	88.3	73.7
INOLML-RN29		87.0	73.1

Unfortunately, unlike other noise benchmark, our method fails to improve the model's performance with bigger size of the validation set (5-10 samples per class). While INOLML still can demonstrate improvement with a validation set of 1 sample per class by a significant margin (3-6%), the model witness a considerable drop in performance when we increase the size of validation set. This phenomenon indicates the

overfitting issues that our method has when we raise the learning speed of the model by expanding the validation set. As our method's selected validation data can contain semantic noise samples (which is prone to overfitting due to semantic noise generation process can produce the noisy labels that can trick the neural networks), this leads to the confirmation biased problem if we utilize such a noisy validation set to guide the model.

3.4.7 Imbalanced Learning

Table 3.8 provides the comparison in performance between our proposed INOLML and several recent meta learning methods when evaluated over imbalanced benchmarks, including MWN [139], FSR [196], FAMUS [171], GDW [18] and L2W [131]. We follow the same setting as FSR [196] since they are one of a few recent meta learning methods without any clean validation data. The results reported in Table 3.8 shows that our INOLML is the current SOTA for meta learning approaches with improvements of 1% to 10% on all benchmarks. Interestingly, the results demonstrate that our proposed informativeness criteria to built the validation set remains effective even when the dataset does not have label noise. This suggests that our selected validation samples are capable of not only addressing noisy data, but also suitably weighting the clean samples according to their informativeness.

3.4.8 Imbalanced Noisy-label Learning

Our proposed INOLML is one of a few methods that addresses both imbalanced learning and noisy-label learning. Following the experimental setting of FSR, we present results on the long-tailed CIFAR datasets with 0.2 to 0.4 symmetric noise rate and

Table 3.8: Test accuracy (%) of our INOLML and other SOTA meta learning approaches evaluated on the CIFAR imbalanced learning (long-tailed) recognition task. The reported results are from Xu et al. [171], Zhang and Pfister [196].

	CIFAR10			**CIFAR100**		
Imb. ratio	**200**	**50**	**10**	**200**	**50**	**10**
Softmax	65.68	74.81	86.39	34.84	43.85	55.71
CB-Focal	65.29	76.71	86.66	32.62	44.32	55.78
CB-Best	68.89	79.27	87.49	36.23	45.32	57.99
L2R	66.51	78.93	85.19	33.38	44.44	53.73
MWN	68.91	80.06	87.84	37.91	46.74	58.46
GDW	-	-	86.8	-	-	56.8
FaMUS	-	83.32	87.90	-	49.93	59.03
FSR-DF	66.15	79.78	88.15	36.74	44.43	55.60
FSR	67.76	79.17	87.40	35.44	42.57	55.45
INOLML	**74.91**	**84.43**	**90.81**	**41.52**	**51.35**	**62.07**

imbalance ratios 10, 50 and 200. Table 3.9 shows the superiority of our INOLML compared to previous methods that attempt to solve this problem, particularly with noise rate 0.4. Unlike CIFAR10, with CIFAR100, we cannot show results with large imbalance ratios (> 10) because that does not leave enough samples for training and validation. For instance, as CIFAR100 contains a total of 50K samples for 100 classes, on average each class has 500 samples. An imbalance ratio of 50 or above will reduce the availability of the minority class to just 10 samples, which is not enough for the extraction of the training data and the validation set, especially when it contains symmetric noise. Nevertheless, based on the results in Table 3.9, on both CIFAR100 benchmarks, our method shows substantially better results compared to previous SOTA methods. Therefore, INOLML has become the new SOTA in most benchmarks for the imbalanced and noisy label learning problem with Resnet32.

Table 3.9: Test accuracy (%) of INOLML and other SOTA methods on CIFAR10 and CIFAR100 imbalanced learning mixed with symmetric noise. The reported results are from [196] and [162].

Dataset	**CIFAR10**						**CIFAR100**	
Noise ratio	**0.2**			**0.4**			**0.2**	**0.4**
Imb. ratio	**10**	**50**	**200**	**10**	**50**	**200**	**10**	**10**
CRUST	65.7	41.5	34.3	59.5	32.4	28.8	-	-
LDAM	82.4	-	-	71.4	-	-	48.1	36.7
LDAM-DRW	83.7	-	-	74.9	-	-	50.4	39.4
BBN	80.4	-	-	70.0	-	-	47.9	35.2
HAR-DRW	82.4	-	-	77.4	-	-	46.2	37.4
ROLT-DRW	85.5	-	-	82.0	-	-	52.4	46.3
FSR	85.7	77.4	65.5	81.6	69.8	49.5	-	-
INOLML	**90.1**	**80.1**	**66.6**	**89.1**	**78.1**	**61.6**	**59.8**	**56.1**

3.4.9 Open-set Noise

Unlike other types of noise, the open-set noise refers to the presence of training images that do not belong to any of the training classes. Following [89], we try our method on three benchmarks that artificially create open set noise in CIFAR10 by mixing into it samples from CIFAR100 and ImageNet. For comparison, we include the results of all the methods in the Distill paper [197] and other meta learning methods [125, 131, 196]. The results are shown in Table 3.10. In general, INOLML shows the best results in all cases, being around 0.5-1% higher when compared with Distill. This is a remarkable result, considering the contamination of the validation set $\mathcal{D}^{(v)}$ (up to 40%) as training progresses, which is shown in Fig. 3.9b. Note that the increase in noise rate in the validation set during training is reasonable because our approach prioritizes informative samples over clean samples to form the validation set. Accordingly, at the start of the training, the framework selects clean samples for the validation set as they have high similarity with samples from the same classes, combined with their high

Table 3.10: Test accuracy (%) of INOLML and previous methods in open-set noise using WideResnet28-10 with 10 samples per class for validation.

Method	ImageNet	CIFAR100	BOTH
RoG [125]	83.4	87.1	84.4
L2R [131]	81.8	81.8	85.0
Distill [197]	94.0	92.3	93.0
INOLML	$\mathbf{94.5 \pm 0.1}$	$\mathbf{93.6 \pm 0.0}$	$\mathbf{93.6 \pm 0.1}$

gradient magnitude because the model is still not well-trained. However, as training progresses, the model starts to converge and the gradient magnitude of clean samples gets smaller, so the algorithm starts to identify "harder" samples, leading to a decrease in the accuracy of our validation set. Unexpectedly, the results for open set noise contrasts with our previous analysis for 0.8 symmetric noise, in which the degradation in performance occurs when just 30% of the validation set is noisy (see Table 3.1), while for open set noise, INOLML still fares well with around 40% noise rate in the validation set. This performance suggests that since noisy samples in open-set noisy-label datasets belong to unknown classes, a validation set with open set noise has lower risk of memorizing the label noise patterns compared to other types of noise, such as symmetric noise. In fact, such open-set noise can have a regularization effect, leading to an overall better performance.

3.4.10 Real-world Datasets

Finally, we present the results of our INOLML on real-world datasets and compare its performance with other other SOTA approaches. All of the compared methods, excluding HAR [16] that uses InceptionResnetV2, report the results using Resnet50 on Webvision benchmark in Table 3.11. Additionally, the results of four different noise

Table 3.11: Prediction accuracy (%) on real-world datasets. Webvision with Resnet50, evaluated on Webvision and ImageNet test sets. The results of other methods are from [26, 196] or from original papers.

Method	Webvision		ImageNet	
	top-1	top-5	top-1	top-5
HAR	75.5	90.7	57.4	82.4
D2L	62.7	84.0	57.8	81.4
Co-teaching	63.6	85.2	61.5	84.7
Iterative-CV	65.2	85.3	61.6	85.0
MentorNet	63.0	81.4	63.8	85.8
CRUST	72.4	89.6	67.4	87.8
GJS	78.0	90.6	74.4	91.2
MW-Net	74.5	88.9	72.6	88.8
FSR	74.9	88.2	72.3	87.2
INOLML	**81.7**	**93.8**	**78.1**	**92.9**

rates evaluated on CNWL dataset (Red Mini-ImageNet) is also provided in Table 3.12. In both cases our methods once again demonstrate the SOTA performance, on Webvision and most of Red Mini-ImageNet settings. For Webvision, we achieved considerable improvement by at least 6% for top-1 accuracy and around 5% improvement on top-5 benchmarks, while for CNWL dataset our framework demonstrates equivalent performance as MOIT model, as well as a clean margin by at least 2% when compared to other benchmarks. Considering that we only use a single PreAct Resnet18 model with meta learning, our INOLML provides another advantage in terms of memory footprint compared to Co-training methods such as DivideMix [92], Co-teaching [51].

3.5 Discussion

In this section, we take a closer look into the optimization in (3.7) and discuss the ablation results. In our optimization in (3.7), our lower-level objective finds validation

Table 3.12: Prediction accuracy (%) on the Red Mini-ImageNet dataset. The results of other methods are from [26, 196] or from original papers.

Method	Noise ratio			
	0.2	**0.4**	**0.6**	**0.8**
CE	47.36	42.70	37.30	29.76
Mix Up	49.10	46.40	40.58	33.58
DivideMix	50.96	46.72	43.14	34.50
MentorMix	51.02	47.14	43.80	33.46
PropMix	61.24	56.22	52.84	43.42
MOIT	63.14	**60.78**	-	**45.88**
FaMUS	51.42	48.06	45.10	35.50
INOLML	**63.23**	58.21	**53.39**	**45.32**

samples that can maximize the training sample weights in (3.6), followed by determining from them samples that can maximize the maximum shared information with the training samples. While the first step is intuitive since it optimizes the same target as the meta learning framework, the second step's effectiveness remains questionable despite our clarification about its necessity in Section 3.3. Hence, in Table 3.13, we put forward an ablation study that analyses this factor with an experiment that replaces Info(.) in (3.7) with the **average of weight** in (3.6). The role and influence of Clean(.) in (3.7) is also explored by another experiment that leaves out the upper-level objective of (3.7) and only optimizes the lower-level objective. We conduct these two experiments on CIFAR10 and CIFAR100 under 0.4 asymmetric noise and 0.2 symmetric noise, combined with imbalanced ratios of 10,20 and 50. Based on the result, each proposed components plays a nontrivial role in improving the model's performance. Naively selecting samples that maximize (3.6) will result in a validation set with near duplicate information as they optimize the same objective. Therefore, our proposed second step addresses this problem by refining the selected samples for the validation

Table 3.13: Test accuracy (%) on CIFAR10 and CIFAR100 under asymmetric and imbalanced noisy-label problems, where IR denotes the imbalance ratio. The 1^{st} row shows the results of the optimization of the average of weight (col. **Average Weight in** (3.6)) instead of (3.7). The 2^{nd} row shows the results of optimizing the lower part of (3.7) (col. Info(.) **Only**) without the upper part of (3.7) Clean(.). The last row (**Whole** (3.7)) shows our final model result.

Average Weight in (3.6)	Info(.) **Only**	**Equation** (3.7)	**0.4 Asymmetric** CIFAR10 WRN IR=1	**0.4 Asymmetric** CIFAR10 RN29 IR=1	**0.2 Symmetric** CIFAR10 RN32 IR=10	**0.2 Symmetric** CIFAR10 RN32 IR=50	**0.2 Symmetric** CIFAR10 RN32 IR=200
✓			91.0	56.6	68.8	37.6	23.4
	✓		92.1	89.3	89.0	79.1	65.9
		✓	94.1	89.8	90.1	80.1	66.6

set to be more diverse, such that every training sample gets benefits from at least a single validation sample of same class. Furthermore, samples that optimize (3.6) have a high risk to be noisy since they have high gradient magnitude. This can lead to confirmation bias if the model overfits such noisy data in the validation set. Hence, Clean(.) is proposed to limit the noise in the clean validation set, and has proven to be effective by raising the performance by 0.7% to 2% in Table 3.13.

Subsequently, we compare the utility of INOLML's validation set, compared with forming the validation set using random sampling and using samples with the highest classification confidence. The results are presented in Fig. 3.10. Our method displays significantly better test accuracy compared to the other two heuristics for building the validation set, where selecting validation samples based on their confidence score produces the worst result, as that leads to a clean but biased set of samples with near duplicate information.

The next analysis focuses on influence of letting the model fit past validation samples. Traditionally, previous meta learning approaches always keep the training set

Table 3.14: Comparison in result of INOLML between using full training data and using a separate subset of training data for validation data sampling under symmetric noise with ratio 0.4 and 0.8 over CIFAR10 and CIFAR100 dataset

Method	**Accuracy**			
	CIFAR10		**CIFAR100**	
	0.4	**0.8**	**0.4**	**0.8**
Full data	91.0	87.9	66.3	59.2
Separate train/val	89.0	83.6	63.1	56.3

and the validation set separate, while our method iteratively extracts part of the training set as the new validation set. While previous works, e.g., [131], argue that this non-separation of the training and validation sets can aggravate the confirmation bias problem during training, we run an experiment to test this issue. We design this experiment by modifying our INOLML, with a division of the noisy training set into two separate training and candidate validation sets at the start of the training, followed by a periodic extraction of the validation set from the initial separate validation set. The experiments show a drop of 2% in accuracy, on average, using such separate validation set, displayed in Table 3.14. The reason for this decrease in performance can be explained by the reduction in the number of training samples, as well as the limit in potential candidates for validation samples.

Additionally, since our INOLML is proposed with the goal of maximizing ω for clean and informative training samples, we conduct further analysis to track the change in the weight ω as training progresses, with and without our methods. Fig. 3.11 demonstrates the tendency of the mean value of ω (labeled as mean weight in the graph) for the clean samples (the left chart) and the noisy ones (the right chart), using CIFAR-10 dataset with 0.8 symmetric noise. From the chart, it is noticeable that our INOLML (blue curve) generates higher overall weight for clean data and lower weight for noisy data,

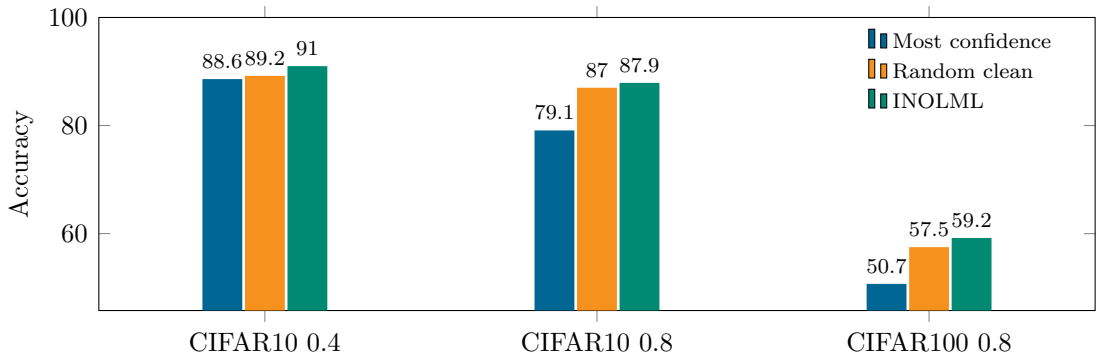
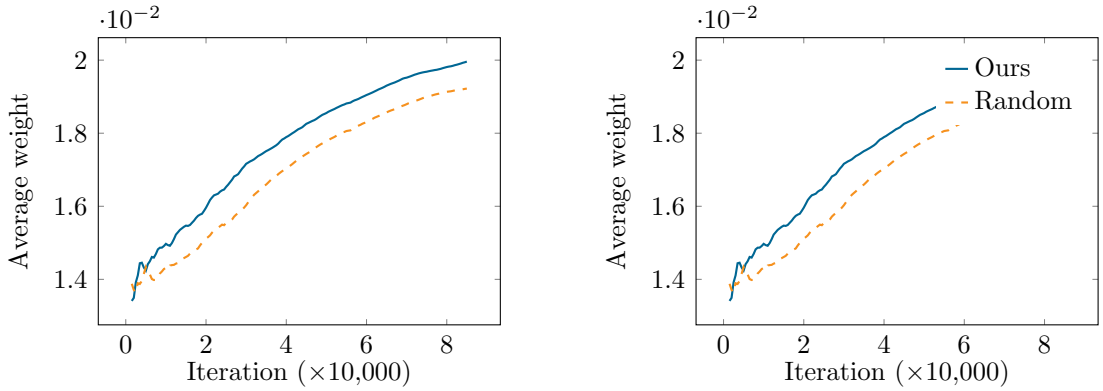


Figure 3.10: Accuracy (%) of our INOLML using different sample selection methods under uniform label noises.



(a) Clean samples

(b) Noisy samples

Figure 3.11: Mean ω given the training iterations, for the selected clean (left side) and noisy (right side) samples using our validation set selection (blue curves) compared to a random validation selection (orange curves) on CIFAR-10 with 0.8 symmetric noise.

compared to classic meta learning framework which randomly selects validation samples (orange curve), thus empirically proving the usefulness of our utility.

The last discussion point is the time efficiency of the proposed INOLML method. For the experiment with CIFAR10, contaminated with symmetric noise, the Distill model takes around 5 and 29 hours for training the Resnet29 and WideResnet28-10 models, respectively. On the other hand, the combination of our method and the Distill requires a slightly higher training time of approximately 5.5 hours for Renset29 model and 31 hours for WideResnet28-10. In summary, INOLML generates an running

time overhead of around 10% to run the validation set optimization at the start of each epoch. All the experiments are conducted on a single NVIDIA V100 GPU. In addition, we also provide the execution time of several recent noisy label learning models. For fair comparison with those approaches, we also run our INOLML using the same PreAct Resnet18 model on 0.4 uniform noise CIFAR100 dataset, using a single V100 GPU. Overall, DivideMix [92] requires around 8.25 hours to finish, CausalNL[175] takes around 12.5h while MOIT [122] needs 8h. Our methods's training lasts around 10.5h, suggesting a competitive running time, compared with other approaches.

3.6 Conclusions

In this work, we have proposed a novel meta learning method, called INOLML, that improves the efficacy of traditional meta learning methods by automatically and progressively producing a pseudo clean validation set from the whole noisy training set. The selection mechanism is based on a new utility criteria that exploits feature and gradient similarities between training and validation samples that takes into account three major factors: sample informativeness, class distribution balance, and label cleanliness. Compared to prior meta learning frameworks, our proposed algorithm does not require a clean validation set, like traditional models such as L2W [131]. INOLML has shown to be more robust to high label noise rates and to achieve SOTA performance on several synthetic and real-world benchmarks. In particular, we set the new SOTA performance in Webvision, CNWL both with real-world noise), and CIFAR10/100 (with synthetic noise), including open-set noise, long-tailed and symmetric label noise. However despite above advantages, our method still has several weaknesses. Firstly, since it is a single model framework, it can suffer from confirmation bias due to the presence of

noisy samples in the validation set. In order to tackle this problem, we plan to integrate the co-training framework with our meta learning in the future.

Additionally, the greedy solution for our bi-label optimization to generate the validation set in (3.7) does not guarantee that the validation set is optimal. Also, the bi-level optimization increases the complexity of the algorithm. Therefore, the proposed methods can still be improved in two ways: we can reduce the model complexity by simplifying the bi-level optimization to a single-level optimization, or we can also replace the greedy strategy with a relaxation method. Moreover, the validation set optimization that happens once per epoch, may not be good enough to keep the validation set up to date with the model's status. Ideally, we should be able to address this problem by updating it more frequently, as long as we can do this at a relatively low computational cost. Lastly, we still lack a theoretical analysis for INOLML utility's influence over the meta learning model's performance. Such analysis will be explored in the future.

Chapter 4

Conclusion and Discussion

This thesis has introduced a novel meta learning algorithm designed to address noisy-label and imbalanced learning problems. My main contribution in the implementation of this new algorithm was a new optimization to automatically build validation sets for meta learning approaches based on innovative criteria that consist of label cleanliness, sample informativeness, and balanced distribution for the validation samples.

More specifically, I proposed a novel validation set optimization to strengthen the effectiveness of the validation set for the meta learning algorithm designed for noisy-label and imbalanced learning problems. The validation set optimization worked by selecting validation samples that will maximize the weights of the training set for the meta learning method. The motivation for this optimization stemmed from theoretical observations about the factors that contribute for the informativeness of validation data in the meta learning framework. I also observed that optimizing only for informativeness can reduce sample diversity in the validation set. Thus, I included a term to increase diversity and another term to balance class distribution to improve robustness to imbalanced learning. In contrast, previous meta learning approaches did not take

into account any of these points and randomly select samples to be manually annotated and included in a validation set. As shown in this thesis, such random selection of validation samples is linked to poor training outcomes and potentially high annotation costs, depending on the number of classes and validation samples per class. Results on imbalanced noisy-label benchmarks demonstrated that the proposed model showed remarkable and promising results over previous meta learning methods, noisy-label learning methods and imbalanced learning methods.

4.1 Limitations and future work

There are many important points that remain to be studied for the method proposed in this thesis. The first one is the optimization of the validation set, which although showing to be effective in general, it is unable to differentiate "hard" clean samples from noisy-label samples because both sample types share similar similarity and gradient magnitude values. Such differentiation would be important because the treatment of these two types of samples should be different during the training process. For instance, noisy-label samples should never be allowed to be inserted into the validation set because they will cause confirmation bias to the training. However, "hard" clean samples should be included into the validation set since it would allow hard training samples to be successfully used for training. Despite the effort to mitigate this problem by periodical renovation of the validation set, I did not manage to solve it, particularly for asymmetric noise. Consequently, my future work in meta learning optimization will aim to study the essential characteristics of "hard" clean samples, and exploit them to optimize further the validation set.

Additionally, the proposed validation data selection does not explicitly explore any

type of feature learning that could, in principle, facilitate the optimization of the validation set. Under real-world noisy-label problems, noisy-label samples often share similar features with other samples from the noisy class, making their feature representation indistinguishable compared to the clean samples from the same noisy class. By exploring feature learning with for example, self-supervised training approaches, the features between noisy-label and clean-label samples could be separated.

Another issue with the proposed approach is the large computational cost involved in the meta learning optimization, which is compounded by the large computational cost of the validation set optimization. As the result, I plan to address this computational cost issue by first simplifying the validation set optimization, and then the meta learning optimization, with the goal to reduce the costs without affecting classification accuracy. A possible solution to be explored is the first-order simplification of the meta learning optimization, such as the one proposed in [196].

The last limitation I plan to address, is the dependence on the available training set to form the validation set. This dependence assumes that we always have enough clean training samples available to build the validation set. However, under high noise rates and low number of training samples per classes, this assumption may not hold. Our future plan is to mitigate this issue with the development of data generation techniques to produce informative validation samples.

Bibliography

[1] G. Algan and I. Ulusoy. Meta soft label generation for noisy labels. *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 7142–7148, 2021.

[2] G. Algan, ilkay Ulusoy, Şaban Gönül, Banu Turgut, and B. Bakbak. Deep learning from small amount of medical data with noisy labels: A meta-learning approach. *ArXiv*, abs/2010.06939, 2020.

[3] D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.

[4] Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2019.

[5] Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *ArXiv*, abs/1701.07875, 2017.

[6] D. Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron C. Courville, Yoshua Bengio, and S. Lacoste-Julien. A closer look at memorization in deep networks. *ArXiv*, abs/1706.05394, 2017.

[7] J. T. Ash, Chicheng Zhang, A. Krishnamurthy, J. Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *ArXiv*, abs/1906.03671, 2020.

[8] HeeSun Bae, Seung-Jae Shin, Joonho Jang, Byeonghu Na, Kyungwoo Song, and Il-Chul Moon. From noisy prediction to true label: Noisy prediction calibration via generative model. In *ICML*, 2022.

[9] William H. Beluch, Tim Genewein, A. Nürnberger, and Jan M. Köhler. The power of ensembles for active learning in image classification. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9368–9377, 2018.

[10] Abhijit Bendale and T. Boult. Towards open set deep networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1563–1572, 2016.

[11] Shruti Bhargava and David A. Forsyth. Exposing and correcting the gender bias in image captioning datasets and models. *ArXiv*, abs/1912.00578, 2019.

[12] Mustafa Bilgic and Lise Getoor. Link-based active learning. In *NIPS Workshop on Analyzing Networks and Learning with Graphs*, 2009. URL http://www.cs.iit.edu/~ml/pdfs/bilgic-nips09-wkshp.pdf.

[13] Avrim Blum and Tom M. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT' 98*, 1998.

[14] Robert Böhm, Cornelia Betsch, Yana Litovsky, Philipp Sprengholz, Noel T. Brewer, Gretchen Chapman, Julie Leask, George Loewenstein, Martha Scherzer, Cass R. Sunstein, and Michael Kirchler. Crowdsourcing interventions to promote

uptake of covid-19 booster vaccines. *eClinicalMedicine*, 53:101632, 2022. ISSN 2589-5370. doi: https://doi.org/10.1016/j.eclinm.2022.101632. URL https://www.sciencedirect.com/science/article/pii/S2589537022003625.

[15] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Aréchiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, 2019.

[16] Kaidi Cao, Yining Chen, Junwei Lu, Nikos Aréchiga, Adrien Gaidon, and Tengyu Ma. Heteroskedastic and imbalanced deep learning with adaptive regularization. In *International Conference on Learning Representations*, 2021.

[17] Jonathan H. Chan and Chenqi Li. Learning from imbalanced covid-19 chest x-ray (cxr) medical imaging data. *Methods*, 202:31–39, 2022. ISSN 1046-2023. doi: https://doi.org/10.1016/j.ymeth.2021.06.002. URL https://www.sciencedirect.com/science/article/pii/S1046202321001547. Machine Learning Methods for Bio-Medical Image and Signal Processing: Recent Advances.

[18] Can Chen, Shuhao Zheng, Xi Chen, Erqun Dong, Xue Liu, Hao Liu, and Dejing Dou. Generalized data weighting via class-level gradient manipulation. In *Advances in Neural Information Processing Systems*, 2021.

[19] Pengfei Chen, B. Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *ICML*, 2019.

[20] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR, 2020.

[21] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E. Hinton. Big self-supervised models are strong semi-supervised learners. *ArXiv*, abs/2006.10029, 2020.

[22] Hao Cheng, Zhaowei Zhu, Xingyu Li, Yifei Gong, Xing Sun, and Yang Liu. Learning with instance-dependent label noise: A sample sieve approach. *ArXiv*, abs/2010.02347, 2020.

[23] Hsin-Ping Chou, Shih-Chieh Chang, Jia-Yu Pan, Wei Wei, and Da-Cheng Juan. Remix: Rebalanced mixup. In *European Conference on Computer Vision*, 2020.

[24] Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling. Feature space augmentation for long-tailed data. In *European Conference on Computer Vision*, 2020.

[25] F. Cordeiro and G. Carneiro. A survey on deep learning with noisy labels: How to train your model when you cannot trust on the annotations? *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 9–16, 2020.

[26] Filipe R. Cordeiro, Vasileios Belagiannis, Ian D. Reid, and G. Carneiro. Propmix: Hard sample filtering and proportional mixup for learning with noisy labels. In *British Machine and Vision Conference*, 2021.

[27] Jiequan Cui, Zhisheng Zhong, Shu Liu, Bei Yu, and Jiaya Jia. Parametric contrastive learning. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 695–704, 2021.

[28] Yin Cui, Yang Song, Chen Sun, Andrew G. Howard, and Serge J. Belongie. Large scale fine-grained categorization and domain-specific transfer learning.

2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4109–4118, 2018.

[29] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge J. Belongie. Class-balanced loss based on effective number of samples. In *Conference on Computer Vision and Pattern Recognition*, pages 9260–9269, 2019.

[30] Ido Dagan and S. Argamon. Committee-based sampling for training probabilistic classifiers. In *International Conference on Machine Learning*, 1995.

[31] M. Dehghani, Aliaksei Severyn, Sascha Rothe, and J. Kamps. Avoiding your teacher's mistakes: Training neural networks with controlled weak supervision. *ArXiv*, abs/1711.00313, 2017.

[32] M. Dehghani, Aliaksei Severyn, Sascha Rothe, and J. Kamps. Learning to learn from weak supervision by full supervision. *ArXiv*, abs/1711.11383, 2017.

[33] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[34] Terrance Devries and Graham W. Taylor. Dataset augmentation in feature space. *ArXiv*, abs/1702.05538, 2017.

[35] Qi Dong, Shaogang Gong, and Xiatian Zhu. Class rectification hard mining for imbalanced deep learning. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1869–1878, 2017.

[36] Charles Peter Elkan. The foundations of cost-sensitive learning. In *International Joint Conference on Artificial Intelligence*, 2001.

[37] Erik Englesson and Hossein Azizpour. Generalized jensen-shannon divergence loss for learning with noisy labels. In *Advances in Neural Information Processing Systems*, 2021.

[38] William Fedus, Mihaela Rosca, Balaji Lakshminarayanan, Andrew M. Dai, Shakir Mohamed, and Ian J. Goodfellow. Many paths to equilibrium: Gans do not need to decrease a divergence at every step. *ArXiv*, abs/1710.08446, 2017.

[39] Chengjian Feng, Yujie Zhong, and Weilin Huang. Exploring classification equilibrium in long-tailed object detection. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3397–3406, 2021.

[40] Alexander Freytag, Erik Rodner, and Joachim Denzler. Selecting influential examples: Active learning with expected model output changes. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 562–577, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10593-2.

[41] Aritra Ghosh, Himanshu Kumar, and P. Sastry. Robust loss functions under label noise for deep neural networks. In *AAAI Conference on Artificial Intelligence*, 2017.

[42] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *ArXiv*, abs/1803.07728, 2018.

[43] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[44] Jianping Gou, B. Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. *ArXiv*, abs/2006.05525, 2020.

[45] Haojie Guo and Song Wang. Long-tailed multi-label visual recognition by collaborative training on uniform and re-balanced samplings. In *Conference on Computer Vision and Pattern Recognition*, pages 15084–15093, 2021.

[46] Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R Scott, and Dinglong Huang. Curriculumnet: Weakly supervised learning from large-scale web images. In *European Conference on Computer Vision*, pages 135–150, 2018.

[47] Yuhong Guo. Active instance sampling via matrix partition. In *NIPS*, 2010.

[48] Deepak Kumar Gupta, Kush Attal, and Dina Demner-Fushman. A dataset for medical instructional video classification and question answering. *ArXiv*, abs/2201.12888, 2022.

[49] Raia Hadsell, Sumit Chopra, and Yann Lecun. Dimensionality reduction by learning an invariant mapping. In *Conference on Computer Vision and Pattern Recognition*, pages 1735 – 1742, 02 2006.

[50] Bo Han, Gang Niu, Jiangchao Yao, Xingrui Yu, Miao Xu, Ivor Tsang, and Masashi Sugiyama. Pumpout: A meta approach for robustly training deep neural networks with noisy labels, 09 2018.

[51] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, M. Xu, Weihua Hu, I. Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 2018.

[52] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems*, pages 8527–8537, 2018.

[53] Bo Han, Gang Niu, Jiangchao Yao, Xingrui Yu, Miao Xu, Ivor Tsang, and Masashi Sugiyama. Pumpout: A meta approach for robustly training deep neural networks with noisy labels. In *ICML Workshop on Uncertainty and Robustness in Deep Learning*, 2019.

[54] H. Harutyunyan, Kyle Reing, G. V. Steeg, and A. Galstyan. Improving generalization by controlling label-noise information in neural network weights. *ArXiv*, abs/2002.07933, 2020.

[55] Alexander Hauptmann, Wei-Hao Lin, Rong Yan, Jun Yang, and Ming yu Chen. Extreme video retrieval: joint maximization of human and computer performance. In *ACM Multimedia*, 2006.

[56] Kaiming He, Ross B. Girshick, and Piotr Dollár. Rethinking imagenet pre-training. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4917–4926, 2018.

[57] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[58] Ruifei He, Jihan Yang, and Xiaojuan Qi. Re-distributing biased pseudo labels for semi-supervised semantic segmentation: A baseline investigation. *2021*

IEEE/CVF International Conference on Computer Vision (ICCV), pages 6910–6920, 2021.

[59] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. In *Advances in Neural Information Processing Systems*, pages 10456–10465, 2018.

[60] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. *ArXiv*, abs/1901.09960, 2019.

[61] Danula Hettiachchi, Vassilis Kostakos, and Jorge Gonçalves. A survey on task assignment in crowdsourcing. *ACM Computing Surveys (CSUR)*, 55:1 – 35, 2021.

[62] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015.

[63] Youngkyu Hong, Seungju Han, Kwanghee Choi, Seokjun Seo, Beomsu Kim, and Buru Chang. Disentangling label distribution for long-tailed visual recognition. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6622–6632, 2020.

[64] Timothy M Hospedales, Antreas Antoniou, Paul Micaelli, and Amos J. Storkey. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[65] N. Houlsby, Ferenc Huszár, Zoubin Ghahramani, and M. Lengyel. Bayesian active learning for classification and preference learning. *ArXiv*, abs/1112.5745, 2011.

[66] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep

representation for imbalanced classification. In *Conference on Computer Vision and Pattern Recognition*, pages 5375–5384, 2016.

[67] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Label propagation for deep semi-supervised learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5065–5074, 2019.

[68] Lee Jaehwan, Yoo Donggeun, and Kim Hyo-Eun. Photometric transformer networks and label adjustment for breast density prediction. In *International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[69] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *ArXiv*, abs/2011.00362, 2020.

[70] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, 2018.

[71] Lu Jiang, Di Huang, Mason Liu, and Weilong Yang. Beyond synthetic noise: Deep learning on controlled noisy labels. In *International Conference on Machine Learning*, 2020.

[72] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*, 2020.

[73] Bingyi Kang, Yu Li, Sai Nan Xie, Zehuan Yuan, and Jiashi Feng. Exploring

balanced feature spaces for representation learning. In *International Conference on Learning Representations*, 2021.

[74] Bingyi Kang, Yu Li, Sai Nan Xie, Zehuan Yuan, and Jiashi Feng. Exploring balanced feature spaces for representation learning. In *International Conference on Learning Representations*, 2021.

[75] Shyamgopal Karthik, Jérôme Revaud, and Chidlovskii Boris. Learning from long-tailed data with noisy labels. *ArXiv*, abs/2108.11096, 2021.

[76] Parneet Kaur, Karan Sikka, and Ajay Divakaran. Combining weakly and webly supervised learning for classifying food images. *ArXiv*, abs/1712.08730, 2017.

[77] Mohammad Khalilia, Sounak Chakraborty, and Mihail Popescu. Predicting disease risks from highly imbalanced data using random forest. *BMC Medical Informatics and Decision Making*, 11:51 – 51, 2011.

[78] Chris Dongjoo Kim, Jinseo Jeong, and Gunhee Kim. Imbalanced continual learning with partitioning reservoir sampling. *ArXiv*, abs/2009.03632, 2020.

[79] Dongha Kim, Yongchan Choi, Kunwoong Kim, and Yongdai Kim. Inn: A method identifying clean-annotated samples via consistency effect in deep neural networks. *ArXiv*, abs/2106.15185, 2021.

[80] Taehyeon Kim, Jongwoo Ko, Sangwook Cho, Ji Sub Choi, and Se-Young Yun. Fine samples for learning with noisy labels. In *Neural Information Processing Systems*, 2021.

[81] Youngdong Kim, Junho Yim, Juseung Yun, and Junmo Kim. NLNL: Negative

learning for noisy labels. In *International Conference on Computer Vision*, pages 101–110, 10 2019.

[82] Youngdong Kim, Junho Yim, Juseung Yun, and Junmo Kim. Nlnl: Negative learning for noisy labels. In *International Conference on Computer Vision*, pages 101–110, 10 2019.

[83] R. King, Ken E. Whelan, F. M. Jones, Philip G. K. Reiser, Christopher H. Bryant, S. Muggleton, D. Kell, and S. Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427:247–252, 2004.

[84] Andreas Kirsch, Joost R. van Amersfoort, and Y. Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In *NeurIPS*, 2019.

[85] Tomohiko Konno and Michiaki Iwazume. Icing on the cake: An easy and quick post-learnig method you can try after deep learning. *ArXiv*, abs/1807.06540, 2018.

[86] A. Krizhevsky. Learning multiple layers of features from tiny images, 2009.

[87] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[88] Jaehwan Lee, Donggeon Yoo, J. Y. Huh, and Hyo-Eun Kim. Photometric transformer networks and label adjustment for breast density prediction. *ArXiv*, abs/1905.02906, 2019.

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

[90] Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. Cleannet: Transfer learning for scalable image classifier training with label noise. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5447–5456, 2018.

[91] D. Lewis and W. Gale. A sequential algorithm for training text classifiers. In *SIGIR '94*, 1994.

[92] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*, 2020.

[93] Shuang Li, Kaixiong Gong, Chi Harold Liu, Yulin Wang, Feng Qiao, and Xinjing Cheng. Metasaug: Meta semantic augmentation for long-tailed visual recognition. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5208–5217, 2021.

[94] Wen Li, Limin Wang, Wei Li, E. Agustsson, and L. Gool. Webvision database: Visual learning and understanding from web data. *ArXiv*, abs/1708.02862, 2017.

[95] Xuefeng Li, Tongliang Liu, Bo Han, Gang Niu, and Masashi Sugiyama. Provably end-to-end label-noise learning without anchor points. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6403–6413. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/li21l.html.

[96] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common

objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *European Conference on Computer Vision*, pages 740–755, Cham, 2014. Springer International Publishing.

[97] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017.

[98] Zilong Lin, Yong yu Shi, and Zhi Xue. Idsgan: Generative adversarial networks for attack generation against intrusion detection. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2018.

[99] Bo Liu, Haoxiang Li, Hao Kang, Gang Hua, and Nuno Vasconcelos. Gistnet: a geometric structure transfer network for long-tailed recognition. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8189–8198, 2021.

[100] Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Wenhui Li. Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In *Conference on Computer Vision and Pattern Recognition*, pages 2967–2976, 2020.

[101] Qingshan Liu, Renlong Hang, Huihui Song, and Zhi Li. Learning multiscale deep features for high-resolution satellite image scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, 56:117–126, 2018.

[102] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. In *Advances on Neural Information Processing Systems*, volume 33, pages 20331–20342, 2020.

[103] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38:447–461, 2014.

[104] Yang Liu. Identifiability of label noise transition matrix. *ArXiv*, abs/2202.02016, 2022.

[105] Yang Liu and Hongyi Guo. Peer loss functions: Learning from noisy labels without knowing noise rates. *International Conference on Machine Learning*, 2019.

[106] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2532–2541, 2019.

[107] I. Loshchilov and F. Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv: Learning*, 2017.

[108] Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. In *International Conference on Machine Learning*, pages 3355–3364. PMLR, 2018.

[109] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, S. Erfani, and J. Bailey. Normalized loss functions for deep learning with noisy labels. *ArXiv*, abs/2006.13554, 2020.

[110] Eran Malach and Shai Shalev-Shwartz. Decoupling" when to update" from"

how to update". In *Advances in Neural Information Processing Systems*, pages 960–970, 2017.

[111] Naresh Manwani and P. Sastry. Noise tolerance under risk minimization. *IEEE Transactions on Cybernetics*, 43:1146–1151, 2013.

[112] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2021.

[113] Qiguang Miao, Ying Cao, Ge Xia, Maoguo Gong, Jiachen Liu, and Jianfeng Song. Rboost: Label noise-robust boosting algorithm based on a nonconvex loss function and the numerically stable base learners. *IEEE Transactions on Neural Networks and Learning Systems*, 27(11):2216–2228, 2015.

[114] Qiguang Miao, Ying Cao, Ge Xia, Maoguo Gong, Jiachen Liu, and Jianfeng Song. Rboost: Label noise-robust boosting algorithm based on a nonconvex loss function and the numerically stable base learners. *IEEE Transactions on Neural Networks and Learning Systems*, 27(11):2216–2228, 2016. doi: 10.1109/TNNLS.2015.2475750.

[115] Baharan Mirzasoleiman, Kaidi Cao, and Jure Leskovec. Coresets for robust training of deep neural networks against noisy labels. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 11465–11477. Curran Associates, Inc., 2020.

[116] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. *2016 IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR), pages 2574–2582, 2015.

[117] Robert Moskovitch, Nir Nissim, Dima Stopel, Clint Feher, Roman Englert, and Yuval Elovici. Improving the detection of unknown computer worms activity using active learning. In *Deutsche Jahrestagung für Künstliche Intelligenz*, 2007.

[118] Neel Nanda, Jonathan Uesato, and Sven Gowal. An empirical investigation of learning from biased toxicity labels. *ArXiv*, abs/2110.01577, 2021.

[119] D. Nguyen, Chaithanya Kumar Mummadi, Thi-Phuong-Nhung Ngo, T. Nguyen, Laura Beggel, and T. Brox. Self: Learning to filter noisy labels with self-ensembling. *ArXiv*, abs/1910.01842, 2020.

[120] Tam Nguyen, C Mummadi, T Ngo, L Beggel, and Thomas Brox. Self: learning to filter noisy labels with self-ensembling. In *International Conference on Learning Representations*, 2020.

[121] Diego Ortego, Eric Arazo, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Towards robust learning with different label noise distributions. *arXiv preprint arXiv:1912.08741*, 2019.

[122] Diego Ortego, Eric Arazo, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Multi-objective interpolation training for robustness to label noise. In *Conference on Computer Vision and Pattern Recognition*, pages 6606–6615, 2021.

[123] Wanli Ouyang, Xiaogang Wang, Cong Zhang, and Xiaokang Yang. Factors in finetuning deep model for object detection with long-tail distribution. *2016 IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR), pages 864–873, 2016.

[124] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Gerardo Flores, George H Chen, Tom Pollard, Joyce C Ho, and Tristan Naumann, editors, *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR, 07–08 Apr 2022. URL https://proceedings.mlr.press/v174/pal22a.html.

[125] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2233–2241, 2017. doi: 10.1109/CVPR.2017.240.

[126] H. Permuter, J. M. Francos, and I. Jermyn. A study of gaussian mixture models of color and texture features for image classification and segmentation. *Pattern Recognit.*, 39:695–706, 2006.

[127] Foster J. Provost. Machine learning from imbalanced data sets 101, 2008.

[128] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015.

[129] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015.

[130] Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. *ArXiv*, abs/2007.10740, 2020.

[131] Mengye Ren, Wenyuan Zeng, Binh Yang, and R. Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, 2018.

[132] Pengzhen Ren, Y. Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and X. Wang. A survey of deep active learning. *ArXiv*, abs/2009.00236, 2020.

[133] N. Roy and A. McCallum. Toward optimal active learning through monte carlo estimation of error reduction. In *ICML 2001*, 2001.

[134] Lars Ruthotto and Eldad Haber. An introduction to deep generative modeling. *GAMM-Mitteilungen*, 44, 2021.

[135] Eric Arazo Sanchez, Diego Ortego, P. Albert, N. O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. *ArXiv*, abs/1904.11238, 2019.

[136] Burr Settles. Active learning literature survey. Technical Report 1648, University of Wisconsin–Madison, 2009.

[137] M. Shardlow, Meizhi Ju, Maolin Li, C. O'Reilly, Elisabetta Iavarone, J. McNaught, and S. Ananiadou. A text mining pipeline using active and deep learning aimed at curating information in computational neuroscience. *Neuroinformatics*, 17:391 – 406, 2018.

[138] Weiwei Shi, Yihong Gong, C. Ding, Zhiheng Ma, Xiaoyu Tao, and Nanning Zheng. Transductive semi-supervised deep learning using min-max features. In *European Conference on Computer Vision*, 2018.

[139] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *NeurIPS*, 2019.

[140] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. SELFIE: Refurbishing unclean samples for robust deep learning. In *ICML*, 2019.

[141] Neil Stewart, Jesse Chandler, and Gabriele Paolacci. Crowdsourcing samples in cognitive science. *Trends in Cognitive Sciences*, 21(10):736–748, 2017. ISSN 1364-6613. doi: https://doi.org/10.1016/j.tics.2017.06.007. URL https://www.sciencedirect.com/science/article/pii/S1364661317301316.

[142] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23:828–841, 2017.

[143] Sainbayar Sukhbaatar and Rob Fergus. Learning from noisy labels with deep neural networks. *arXiv preprint arXiv:1406.2080*, 2(3):4, 2014.

[144] Haoliang Sun, Chenhui Guo, Qi Wei, Zhongyi Han, and Yilong Yin. Learning to rectify for robust learning with noisy labels. *Pattern Recognition*, page 108467, 2021.

[145] Yanmin Sun, Mohamed S. Kamel, Andrew Wong, and Yang Wang. Cost-sensitive

boosting for classification of imbalanced data. *Pattern Recognition*, 40:3358–3378, 12 2007.

[146] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. *ArXiv*, abs/2009.12991, 2020.

[147] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, pages 1195–1204, 2017.

[148] Junjiao Tian, Yen-Cheng Liu, Nathan Glaser, Yen-Chang Hsu, and Zsolt Kira. Posterior re-calibration for imbalanced datasets. *ArXiv*, abs/2010.11820, 2020.

[149] Nidhi Vyas, Shreya Saxena, and T. Voice. Learning soft labels via meta learning. *ArXiv*, abs/2009.09496, 2020.

[150] Gefei Wang, Yuling Jiao, Qiang Xu, Yang Wang, and Can Yang. Deep generative learning via schrödinger bridge. In *International Conference on Machine Learning*, 2021.

[151] Jianfeng Wang, Thomas Lukasiewicz, Xiaolin Hu, Jianfei Cai, and Zhenghua Xu. Rsg: A simple but effective module for learning imbalanced datasets. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3783–3792, 2021.

[152] Tao Wang, Yu Li, Bingyi Kang, Junnan Li, Jun Hao Liew, Sheng Tang, Steven C. H. Hoi, and Jiashi Feng. The devil is in classification: A simple framework for

long-tail instance segmentation. In *European Conference on Computer Vision*, 2020.

[153] Tao Wang, Yu Li, Bingyi Kang, Junnan Li, Jun Hao Liew, Sheng Tang, Steven C. H. Hoi, and Jiashi Feng. The devil is in classification: A simple framework for long-tail instance segmentation. In *European Conference on Computer Vision*, 2020.

[154] Xinshao Wang, Yang Hua, Elyor Kodirov, and N. Robertson. Imae for noise-robust learning: Mean absolute error does not treat examples equally and gradient magnitude's variance matters. *arXiv: Learning*, 2019.

[155] Xinshao Wang, Yang Hua, Elyor Kodirov, and Neil M Robertson. IMAE for noise-robust learning: Mean absolute error does not treat examples equally and gradient magnitude's variance matters. *arXiv preprint arXiv:1903.12141*, 2019.

[156] Yiru Wang, Weihao Gan, Wei Wu, and Junjie Yan. Dynamic curriculum learning for imbalanced data classification. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5016–5025, 2019.

[157] Yisen Wang, Weiyang Liu, Xingjun Ma, J. Bailey, H. Zha, Le Song, and Shutao Xia. Iterative learning with open-set noisy labels. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8688–8696, 2018.

[158] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and J. Bailey. Symmetric cross entropy for robust learning with noisy labels. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 322–330, 2019.

[159] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey.

Symmetric cross entropy for robust learning with noisy labels. In *International Conference on Computer Vision*, pages 322–330, 2019.

[160] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Advances on Neural Information Processing Systems*, 2017.

[161] Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Loddon Yuille, and Fan Yang. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10852–10861, 2021.

[162] Tong Wei, Jiang-Xin Shi, Wei-Wei Tu, and Yu-Feng Li. Robust long-tailed learning under label noise. *ArXiv*, abs/2108.11569, 2021.

[163] Sebastien C. Wong, Adam Gatt, Victor Stamatescu, and Mark D. McDonnell. Understanding data augmentation for classification: When to warp? *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–6, 2016.

[164] Tong Wu, Ziwei Liu, Qingqiu Huang, Yu Wang, and Dahua Lin. Adversarial robustness under long-tailed distribution. In *Conference on Computer Vision and Pattern Recognition*, pages 8655–8664, 2021.

[165] Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. Are anchor points really indispensable in label-noise learning? In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/9308b0d6e5898366a4a986bc33f3d3e7-Paper.pdf.

[166] Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. Are anchor points really indispensable in label-noise learning? In *Advances in Neural Information Processing Systems*, pages 6838–6849, 2019.

[167] Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7597–7610. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/5607fe8879e4fd269e88387e8cb30b7e-Paper.pdf.

[168] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Conference on Computer Vision and Pattern Recognition*, pages 2691–2699, 2015.

[169] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans. *ArXiv*, abs/2112.07804, 2021.

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

[171] Youjiang Xu, Linchao Zhu, Lu Jiang, and Yi Yang. Faster meta update strategy for noise-robust deep learning. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[172] Cheng Xue, Qi Dou, Xueying Shi, Hao Chen, and Pheng-Ann Heng. Robust

learning at noisy labeled medical images: Applied to skin lesion classification. In *International Symposium on Biomedical Imaging*, pages 1280–1283, 2019.

[173] Yuzhe Yang and Zhi Xu. Rethinking the value of labels for improving class-imbalanced learning. *ArXiv*, abs/2006.07529, 2020.

[174] Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. Dual t: Reducing estimation error for transition matrix in label-noise learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7260–7271. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/512c5cad6c37edb98ae91c8a76c3a291-Paper.pdf.

[175] Yu Yao, Tongliang Liu, Mingming Gong, Bo Han, Gang Niu, and Kun Zhang. Instance-dependent label-noise learning under a structural causal model. *ArXiv*, abs/2109.02986, 2021.

[176] Kun Yi and Jianxin Wu. Probabilistic end-to-end noise correction for learning with noisy labels. In *Conference on Computer Vision and Pattern Recognition*, pages 7010–7018, 06 2019.

[177] Changchang Yin, Buyue Qian, Shilei Cao, Xiaoyu Li, Jishang Wei, Qinghua Zheng, and Ian Davidson. Deep similarity-based batch mode active learning with exploration-exploitation. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 575–584, 2017. doi: 10.1109/ICDM.2017.67.

[178] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker.

Feature transfer learning for deep face recognition with long-tail data. *ArXiv*, abs/1803.09014, 2018.

[179] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor W Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *International Conference on Machine Learning*, 2019.

[180] Xiyu Yu, Tongliang Liu, Mingming Gong, and D. Tao. Learning with biased complementary labels. *ArXiv*, abs/1711.09535, 2018.

[181] Xiyu Yu, Tongliang Liu, Mingming Gong, and Dacheng Tao. Learning with biased complementary labels. In *European Conference on Computer Vision*, pages 68–83, 2018.

[182] Bodi Yuan, Jianyu Chen, Weidong Zhang, Hung-Shuo Tai, and Sara McMains. Iterative cross learning on noisy labels. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 757–765, 2018. doi: 10.1109/WACV.2018.00088.

[183] Bodi Yuan, Jianyu Chen, Weidong Zhang, Hung-Shuo Tai, and Sara McMains. Iterative cross learning on noisy labels. In *IEEE Winter Conference on Applications of Computer Vision*, pages 757–765, 2018.

[184] Michal Zajac, Konrad Zolna, Negar Rostamzadeh, and Pedro H. O. Pinheiro. Adversarial framing for image and video classification. *ArXiv*, abs/1812.04599, 2018.

[185] Yuhang Zang, Chen Huang, and Chen Change Loy. FASA: Feature augmentation

and sampling adaptation for long-tailed instance segmentation. In *International Conference on Computer Vision*, 2021.

[186] Cheng Zhang, Tai-Yu Pan, Yandong Li, Hexiang Hu, Dong Xuan, Soravit Changpinyo, Boqing Gong, and Wei-Lun Chao. Mosaicos: A simple and effective use of object-centric images for long-tailed object detection. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 407–417, 2021.

[187] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[188] Jing Zhang, V. Sheng, Qianmu Li, Jian Wu, and X. Wu. Consensus algorithms for biased labeling in crowdsourcing. *Inf. Sci.*, 382-383:254–273, 2017.

[189] Weihe Zhang, Yali Wang, and Yu Qiao. Metacleaner: Learning to hallucinate clean representations for noisy-labeled visual recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7365–7374, 2019. doi: 10.1109/CVPR.2019.00755.

[190] Weihe Zhang, Yali Wang, and Yu Qiao. MetaCleaner: Learning to hallucinate clean representations for noisy-labeled visual recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 7373–7382, 2019.

[191] Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao. Range loss for deep face recognition with long-tailed training data. In *International Conference on Computer Vision*, pages 5419–5428, 2017.

[192] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *ArXiv*, abs/2110.04596, 2021.

[193] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *arXiv:2110.04596*, 2021.

[194] Yikai Zhang, Songzhu Zheng, Pengxiang Wu, Mayank Goswami, and Chao Chen. Learning with feature dependent label noise: a progressive approach. *International Conference on Feature Representations*, 2021.

[195] Zhenwei Zhang and Qingyun Du. Hourly mapping of surface air temperature by blending geostationary datasets from the two-satellite system of goes-r series. *ISPRS Journal of Photogrammetry and Remote Sensing*, 183:111–128, 2022. ISSN 0924-2716. doi: https://doi.org/10.1016/j.isprsjprs.2021.10.022. URL https://www.sciencedirect.com/science/article/pii/S0924271621002902.

[196] Zizhao Zhang and Tomas Pfister. Learning fast sample re-weighting without reward data. In *International Conference on Computer Vision*, 2021.

[197] Zizhao Zhang, Han Zhang, Sercan Ö. Arik, Honglak Lee, and Tomas Pfister. Distilling effective supervision from severe label noise. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9291–9300, 2020.

[198] Chen Zhao, Ren Shuai, Li Ma, Wenjia Liu, and Menglin Wu. Improving cervical cancer classification with imbalanced datasets combining taming transformers with t2t-vit. *Multimedia Tools and Applications*, 81:24265 – 24300, 2022.

[199] Manqi Zhao, Shengyang Li, Shiyu Xuan, Longxuan Kou, Shuai Gong, and

Zhuang Zhou. Satsot: A benchmark dataset for satellite video single object tracking. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–11, 2022.

[200] Evgenii Zheltonozhskii, Chaim Baskin, Avi Mendelson, Alex M Bronstein, and Or Litany. Contrast to divide: Self-supervised pre-training for learning with noisy labels. *arXiv preprint arXiv:2103.13646*, 2021.

[201] Yaoyao Zhong, Weihong Deng, Mei Wang, Jiani Hu, Jianteng Peng, Xunqiang Tao, and Yaohai Huang. Unequal-training for deep face recognition with long-tailed noisy data. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7804–7813, 2019.

[202] Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16484–16493, 2020.

[203] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. BBN: bilateral-branch network with cumulative learning for long-tailed visual recognition. *Conference on Computer Vision and Pattern Recognition*, pages 9716–9725, 2020.

[204] Zhi-Hua Zhou and Xu-Ying Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):63–77, 2006.

[205] Linchao Zhu and Yi Yang. Inflated episodic memory with region self-attention for long-tailed visual recognition. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4343–4352, 2020.

[206] Zhaowei Zhu, Yiwen Song, and Yang Liu. Clusterability as an alternative to

anchor points when learning with noisy labels. In *International Conference on Machine Learning*, 2021.

[207] Liu Ziyin, Blair Chen, Ru Wang, Paul Pu Liang, R. Salakhutdinov, Louis-Philippe Morency, and Masahito Ueda. Learning not to learn in the presence of noisy labels. *ArXiv*, abs/2002.06541, 2020.