Challenges and Properties in Deep Learning for Medical Imaging, Types of Noisy Labels, Handling Methods, and Their Limitations

Jason Rich Darmawan
School of Computer Science
Northwestern Polytechnical University
Xi'an, China
jasonrichdarmawan@mail.nwpu.edu.cn

Abstract

Objectives: The performance of deep learning methods can be impacted by label noise. Despite this, existing methods tend to focus on specific aspects of label, addressing only part of the problem while overlooking others. This review aims to identify the challenges, properties associated with label noise, categorize the types of label noise, the handling methods, define the basic concepts of each category, and the limitations inherent to these approaches.

Methods: We searched using Google Scholar. Our search terms include "noisy/noise/uncertainty/uncertain label", "label noise learning".

Results: A total of 95 papers referenced, 7 challenges, 3 properties associated with label noise, 4 types of label noise, and 8 types of handling methods.

Keywords—label noise learning, types of noisy labels

1 Introduction

Manual methods [1] and current automated methods [2] used to create training dataset for medical image also generate label noise [3]. The performance of deep learning methods can be impacted by label noise [3], [4], [5], [6], [7], [8], [9], [10]. However, this issue has not received much attention in studies focused on medical image [3].

There are at least four reviews related to noisy labels. For example, [11] focused on model architecture change to handle label noise [3]. [12] focused on methods to handle label noise [3]. [13] focused on sample selection and loss correction methods to handle label noise [3]. [3] focused on methods to handle label noise in the medical domain.

This study aims to address the following critical research questions within the domain of label noise learning in medical imaging.

1. How to detect label noise?
2. How to handle label noise?
3. How to ensure that the selected parameters generalize beyond the training set?

This paper makes several contributions within the domain of label noise learning in medical imaging, outlined as follows:

1. Identify seven challenges associated with label noise.
2. Identify three properties associated with label noise.
3. Identify four types of label noise.
4. Identify eight categories of methods to detect and handle label noise.
5. Identify robust optimizers as methods to ensure selected parameters generalize beyond the training set.

2 Methods

2.1 Eligibility criteria

The inclusion criteria is English-language articles, including conference papers, journal articles, and preprint articles.

2.2 Information sources

The database used for searching is Google Scholar.

2.3 Search strategy

The following keyword used are: “noisy/noise/uncertainty/uncertain label”, “label noise learning”.

2.4 Study selection

The categorizing strategy is summarizing by the challenges, the properties associated with label noise, the types of label noise and methods to detect and handle label noise.

3 Results

3.1 Summary of included literature

A total of 95 papers were referenced from 1 database.

3.2 Overview of challenges associated with label noise

There are at least 7 challenges associated with label noise:

1. **Scarcity of Expert-Annotated Labels**: Expert-annotated labels are scarce due to the high cost and time required for manual annotation [3]. Consequently, researchers use alternative methods, such as crowdsourcing, Natural Language Processing (NLP), or semi-supervised pseudo-labelling [3]. However, these methods often introduce label noise [3].
2. **Observed Variability**: Observer variability arises from factors such as annotators' fatigue [1], [2] or differences in expertise among annotators [14], [15], [16], [17], [18], [2]. Additionally, diagnosing diseases can be challenging even for experts due to the similarity between skin cancer and benign ones [19], [20], [3].
3. **Noisy Public Datasets**: Many publicly available datasets contain inherent label noise [21], [22].
4. **Variability of Noise Across Labels**: Noise levels vary across different classes [23], [24]. The loss distribution of clean and noisy samples differs between majority and minority classes [25]. For instance, female patients are more likely than male patients to be underdiagnosed with cardiovascular disease during clinical care, leading to misdiagnosis and mislabeling [23].
5. **Class Hardness Problem**: The difficulty of classification varies across classes, posing challenge for removing label noise [3]. For inherently difficult classes, high loss values may be observed early in training [3]. Noise removal strategies based on loss value may inadvertently degrade the classifier performance for these classes by mistakenly discarding clean but challenging samples.
6. **Memorization by Neural Networks**: Neural networks with a large number of parameters are prone to memorizing the entire training set [26].
7. **Batch Size and Loss Landscape**:
   The size of the training batch can significantly influence the behavior of the optimization process. Larger batch sizes increase the likelihood of the network settling into a sharp minimum of the training loss rather than a flatter minimum, which typically generalize better [27]. For deep learning models, the training loss is typically non-convex with respect to model's parameters and can be visualized as a landscape with multiple local and global minima. These minima often yield similar training loss values but differ significantly in their generalization performance [28]. Local minima occurs when the loss function $L(\theta)$ is smaller than or equal to nearby values in the parameter space, while global minima represent the smallest values across all possible parameter configurations.

Neural network optimizers typically begin their search randomly within this landscape. Smaller batch sizes introduce more noise into gradient estimation, which can help the optimizer escape sharp basins [27]. Intuitively, the noise increases the loss value at each step, causing the gradients to take larger steps. This helps push the search process out of sharp basins and guide the optimizer to settle into a flatter minimum, where the noise is not insufficient to force it out of the basin [27].

It is also important to note that the commonly visualized training loss graph, which shows $L(\theta)$ changing over epochs, is a 2D projection of the high-dimensional optimization process. The high dimensionality arises from the large number of parameters (e.g., weights and biases) that define the model. The observed minima correspond to specific points along the optimizer's path but do not necessarily represent all possible minima. This suggests that the optimizer might miss better minima due to the stochastic nature of the training process.

3.3 Overview of properties associated with label noise

There are at least 3 properties associated with label noise:

1. **Latent Feature Similarity**: Nearby samples in the latent feature space tend to share similar labels but may also include noisy samples [3].
2. **Cluster Density in the Feature Space**: The density of clusters, constructed based on the deep feature space, approximates the complexity of samples. The assumption is that clusters with high density consist of easy-to-classify samples, while low-density clusters contain hard-to-classify samples. However, samples with similar features—regardless of their true class—are inherently close together, increasing the likelihood that high-density clusters may also contain noisy samples [29].
3. **Model Agreement on Labels**:
   Two compatible models, trained on independent views, tend to agree on the labels of most samples. Conversely, they are unlikely to agree on an incorrect label [30], [31].
   1. **Compatible Models**: This term refers to models that can learn from each other. The independence of the views ensures that the models do not reinforce each other with redundant information, instead providing unique insights. For instance, one view might be based on color histograms, while the other could rely deep features extracted using a pre-trained model based on the same sample.

   For example, Model A may predict a label with high confidence for an unlabeled sample and add this sample to the labeled set of Model B, as demonstrated in [32], [4]. Alternatively, models can learn from each other through a joint loss function, such

as global contrastive loss, which enables them to learn from and teach each other based on the same sample as shown in [33].

A joint loss function contrasts with methods such as those in [30], [19], where one model simply provides high confidence labeled samples to the model without directly updating its own knowledge. In contrasts, a joint loss function allows both models to benefit and refine their outputs based on mutual feedback.

Global Contrastive Loss:

The contrastive loss proposed in [33] is defined as:

$$L_{\text{con}} = D_{\text{KL}}(p_1||p_2) + D_{\text{KL}}(p_2||p_1),$$

where $D_{\text{KL}}(p_1||p_2)$ represents the Kullback-Leibler (KL) divergence between the predicted probability distributions by the two models. Their loss function is similar to a symmetrized KL divergence [34], [35], [36]

KL Divergence Definition:

The KL divergence is defined as:

$$D_{\text{KL}}(p_1||p_2) = \sum_{i=1}^{N}\sum_{m=1}^{M} p_1^m(x_i)\ln\frac{p_1^m(x_i)}{p_2^m(x_i)},$$

where N is the number of samples, M is the number of classes, and $p_1^m(x_i)$ denotes the predicted probability of instance x_i. This ensures that when both models agree, $D_{\text{KL}}(p_1||p_2) = 0$.

Local Contrastive Loss:

Alternatively, [19] proposed a different perspective by using the global contrastive loss to maximize agreement between the two models in predicting a sample. In addition to the global contrastive loss, and introduced a local contrastive loss designed to learn from noisy samples without relying on their labels. This approach reduces the negative effects of noisy samples.

3.4 Overview of types of label noise

There are at least four types of label noise:

1. **Symmetric Noise**: Noisy labels are generated by randomly flipping instance labels to other classes with equal probability, irrespective of the original class. This process is referred to as symmetric because the probability of flipping from class i to class j is equal to the probability of flipping from class j to class i. Consequently, this approach results in a uniform distribution of noise across all classes, as demonstrated in [37], [38], [39].

   Category limitations:

   a. **Predictability**: The noise is predictable because it is scattered and sparse, making it easier to distinguish within clusters of each class.

2. **Asymmetric Noise or Class Dependent Noise**: Noisy labels are generated by flipping instance labels to other classes according to a predefined noise transition matrix $T(i, j)$, where i and j represents the probability of flipping a label from class i to class j. This matrix is typically asymmetric, meaning that the probability of flipping from class i to class j is not necessarily equal to the probability of flipping from class j to class i, as defined in $[32]$.

   Category limitations:

   a. **Predictability**: The noise is predictable because the noise transition is fixed, making it deterministic once estimated.

3. **Instance dependent noise**: Noisy labels are generated by flipping instance labels to other classes based on a noise transition matrix $T(i, j, X)$, which depends on the features of the instance X. Unlike symmetric or asymmetric noise, this approach assigns a unique probability of label flipping to each individual instance, as demonstrated in [40], [41], [42].

   Properties:

   a. **Ambiguity Sensitivity**: Instances that are positioned near the decision boundary or exhibit ambiguous features are more prone to label flipping, as demonstrated in [43].

   Category limitations:

   a. **Predictability**: The noise is predictable because the loss distributions of clean and noisy samples are statistically distinguishable, as demonstrated in [43].

4. **Loss-dependent noise**: Noisy labels are generated by flipping instance labels to other classes in a way that makes the clean and noisy labels statistically indistinguishable.

   Example:

   a. **BadLabel** [43] proposed to flip instance labels to other classes by first calculating the sample loss value of an instance label when it is flipped to a different class. For instance, for a given sample x_n, where n denotes the sample index, flipping the label from class A to class B may results in a loss of 0.1, while flipping from class A to class C results in a loss of 0.11. Instance labels are then selectively flipped, prioritizing those with the highest loss values. This technique ensures that the loss distributions of clean and noisy labels become indistinguishable, with noisy labels systematically positioned far from the decision boundaries. Consequently, this method makes it difficult for the model to differentiate between clean and noisy labels during training.

3.5 Overview of types of handling methods

We categorize existing methods for handling label noise into eight categories, focusing on sample selection and re-weighting, knowledge distillation, label correction, noise transition matrix estimation, mean and covariance estimation, robust loss functions, and data robust optimizers

1. **Sample Selection and Re-weighting**:
   This training strategy involves identifying clean samples and adjusting their weights in the loss function to emphasize their importance, while diminishing the weights of samples with noisy labels.

Table 3.1 Overview of Sample Selection and Reweighting

Reference	Sample selection based on	Re-weighting based on
[32]	Small-loss samples	-
[44]	Small-loss samples	-
[45]	Labels variance by experts; for ranking	Prediction variance via dropout
[33]	Small-loss samples	-
[46]	Gaussian Mixture Model (GMM) [47]	-
[43]	Bayesian Gaussian Mixture Model (BayesGMM) [48]	-
[20]	Loss value; for ranking	Clusters which use Pre-SoftMax layer as input
[19]	Clusters which use the loss distribution as input and output clean probability of each sample. Two networks (i.e. network A select training samples for network B and vice versa)	Global (i.e. between images) and local (i.e. between two augmented images of the same image) contrastive loss
[18]	(1) static value based on agreement between experts and considers the agreement occur by chance; (2) dynamic value based on prediction variance when dropout is applied. The value is used to rank the samples	Apply per-batch normalization to the prediction probabilities

Notable approaches:

a. **Co-Teaching** [32] proposed to use two networks, where each network selects training samples for the other network by identifying samples with the lowest loss. This approach reduces the risk of reinforcing errors that might occur if each network were trained independently.

Limitations:

i. **Reliance on Loss Values**: The selection of clean samples is based on loss values. In cases where noise is substantial, or the dataset has been compromised by a label-flipping attack—such that the loss distributions of clean and noisy samples become statistically indistinguishable—the networks may struggle to identify genuinely clean samples using the small-loss sample criterion.
ii. **Exclusion of Noisy Samples**: Noisy samples are filtered out rather than being utilized. As a result, the model does not leverage the potentially useful features present in noisy samples, which could otherwise contribute to the learning process.
iii. **Class Hardness Problem**: This approach may perform poorly when faced with the class hardness problem, where certain classes are inherently difficult to classify.

b. **Co-Teaching+** [44] improves upon of Co-Teaching [32] by introducing a mechanism to filter overlapping samples between the training sets proposed by the two networks.

Limitations:

i. **High Noise Rate**: When the dataset has an extremely high noise rate, only a small number of samples will be used for training in each mini-batch, potentially reducing the model's ability to generalize [33].

c. **Dual Uncertainty Estimation** [45] proposed to do sample selection by calculating disagreement and re-weighting at each training iteration by applying random dropout. First, it selects training samples with low disagreement among experts (i.e., samples with less conflicting labels from multiple experts). This is done by calculating their disagreement (i.e. using improvement of Direct Uncertainty Prediction (DUP) [49]). Then, it adjusts the importance of the remaining samples by estimating their uncertainty. This is achieved by performing multiple forward passes through the model with random dropout enabled on the same image and averaging the predictions (i.e. using Monte Carlo (MC) Dropout [50]).

Limitations:

i. **Multiple Expert Annotations Required**: The technique requires multiple annotations from different experts.
i. **Exclusion of Hard Samples**: Samples are filtered out based on the disagreement noise. As a result, the model does not initially learn from hard samples—samples where experts provide conflicting labels.
ii. **Single-target Noise**: The technique introduces single-target noise, which refers to errors caused by majority decisions. For example, if 3

out of 5 experts annotate a sample as class A, but the true class is class B, the majority decision introduces noise. The method proposed in the paper calculates single-target noise as the model's confidence in its predictions when specific layers are randomly dropped. However, in practice, this approach encourages the model to become overconfident in its predictions and does not mitigate the errors caused by majority decisions.

d. **Joint Training with Co-Regularization (JoCoR)** [33] proposed to use two networks with a shared joint loss function, rather than assigning each network its own independent loss function. The assumption is that two models trained with independent views—each provides unique insights and no redundant information—will agree on most correctly labeled samples but are unlikely to agree on incorrect labeled ones. Intuitively, this process can be compared to a real-world scenario where student A teaches student B, while the teacher (student A) also reinforces their own knowledge and learning from student B's mistakes. This knowledge reinforcement mechanism is absent in Co-Teaching [32].

   Limitations:

   i. JoCoR [33] shares limitations similar to Co-Teaching [32]. However, it is important to note that JoCoR [33] does not do sample selection for the other network.

e. **Jiménez-Sánchez et al.** [18] proposed to do sample selection based on domain prior knowledge of a class (i.e. agreement between clinical experts and considers the agreement occur by chance) or model uncertainty in predicting a sample and re-weighting. First, when domain prior knowledge is available, it selects training samples by initializing a static scoring value (i.e. probability of selecting a sample with class t) per class for each task or dataset by computing the Cohen's kappa [51], [52], defined as:

$$\kappa = \frac{p_o - p_c}{1 - p_c},$$

where the observed agreement, defined as:

$$p_0 = \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\mathbb{1}\left(y_1^i = y_2^i = \cdots = y_R^i = t\right)$$

represents the proportion of clinical experts who agree on a class of a sample, and the chance agreement, defined as:

$$p_c = \sum_{t=1}^{T}\prod_{r=1}^{R}\frac{n_{r,t}}{N}$$

is the proportion of agreement expected by chance based on the number of times clinical experts predicted each class t; when domain prior knowledge is unavailable, it ranks and selects samples based on uncertainty estimated from model predictions using MC dropout [50], which calculates prediction variance through multiple stochastic forward passes, similar to [45]. Then, it applies a weight to the loss function where weight, defined as:

$$\hat{\alpha}_{k,j}^{e} = \frac{\hat{p}_{k,j}^{e}}{\max_{k} \hat{p}_{k,j}^{e}}$$

is the per-batch normalized prediction probabilities, k represents the kth sample from the jth batch to the loss function.

f. **DivideMix** [46] proposed to use two networks to do sample selection for the other network by using the loss of each sample as input to a Gaussian Mixture Model (GMM) [48]. The GMM outputs clean probabilities, which are then used to do sample selection. The method further augments the clean and noisy sets using mixup [53], refines the labels of the clean sets and infer the labels for the noisy sets.

   Limitations:

   i. **GMM Assumptions**: The GMM [47] assumes the presence of two distinct loss distributions within the dataset—one for clean samples and one for noisy samples. However, this assumption may fail in scenarios like BadLabel [43], a label-flipping attack that intentionally mimics the clean loss distribution.

g. **Robust DivideMix** [43] proposed to use two networks to do sample selection for the other network by using the loss of each sample as input to a Bayesian Gaussian Mixture Model (BayesGMM) [47]. The BayesGMM [47] outputs clean probabilities, which are then used to do sample selection. The method proceeds in four steps: First, the networks are initialized and trained for few epochs. Second, the training set is perturbed. Third, the BayesGMM is used to do sample selection for the other network. Lastly, during inference, predictions are made using both networks.

h. **Xue et al.** [20] proposed to do sample selection and re-weighting at each iteration. First, it selects training samples with low loss. Then, it checks how close each remaining sample is to its neighbors based on the pre-SoftMax layer feature. If a sample is in a less crowded area (i.e. fewer nearby samples), it is given a smaller weight, making it less important during training (i.e. using probabilistic Local Outlier Factor (LOF) [54]).

i. **Xue et al.** [19] proposed to do sample selection using two networks and re-weighting by adding both global (i.e. between different images) and local (i.e.

between two augmented versions of the same image) contrastive into the loss function. First, training samples with low loss are selected for network A based on the loss values calculated by network B, similar to Co-Teaching [32]. To achieve this, the loss distribution is modeled using a Gaussian Mixture Model (GMM) [47]. Then, a global contrastive loss is applied by averaging the two Kullback-Leibler (KL), which resembles the symmetrized Kullback-Leibler (KL) divergence [34], [35], [36], across all samples. Additionally, a local contrastive loss [55] is applied specifically to the noisy samples to learn from the noisy samples without the label.

j. **Iterative Noisy Cross-Validation (INCV)** [56] proposed to train two networks on different parts of the training set, then uses cross-validation to select clean inputs. INCV uses the network architecture used in Co-teaching [32].

Category limitations:

a. **Dropout Scaling**: Existing methods which use dropout (i.e. randomly zeroes some of the elements of the input with probability p) does not put emphasize to scale the outputs by a factor of $\frac{1}{1-p}$ during training to guarantee the perturbed outputs is in expectation equal to the outputs if it does not use dropout [57].

b. **Threshold Determination**: The difficulty to determine the threshold value used in the sample selection process [58], [59], [4], [60], [61].

c. **Classifier Performance on Hard Classes**: Selecting samples based on their loss value might worsen the classifier's performance for hard-to-classify classes. This happens because samples that are difficult to classify, even when correctly labeled, have high loss values [3]. As a result, the classifier may not learn from these challenging samples during the initial stages of training. To address this, [29] proposed a method that categorizes samples into easy, medium and hard groups based on the local density of each cluster. The cluster is constructed by reducing the dimensionality of deep feature representations using the t-distributed Stochastic Neighbor Embedding (t-SNE) [62], followed by clustering the results with the K-means algorithm [63]. Additionally, [25] introduced a method to minimize the adverse effects of imbalanced distribution by adding a term into the loss function that enforces invariance across distributions.

d. **Assumption Validity**: The assumption that two models trained on independent views will mostly agree on correct labels and are unlikely to agree on incorrect labels lacks mathematical proof to guarantee the method's

effectiveness under certain conditions. Consequently, such methods may not consistently achieve stable performance across all real-world datasets [64].

e. **The collaborative learning limitations:** When a class contains a large number of noisy samples, the critical statistics of the class can become dominated by these noisy samples. Consequently, network A may inaccurately predict many noisy samples as clean samples and inadvertently select them for network B to train on.

2. **Knowledge distillation**: A training strategy in which a student model learns from the predictions of the teacher model. Over time, the student model evolves to take on the role of the teacher model, typically by averaging the student models across multiple epochs or checkpoints, and subsequently generates predictions to guide the training of future student models, as demonstrated in Robust Stochastic Knowledge Distillation (RoS-KD) [15].
    a. **RoS-KD** [15] proposed to distill knowledge from multiple teacher networks (different model architectures) and assigns random weights to teacher networks during each iteration, ensuring that only one teacher has a strong influence at a time. First, it trains the teacher networks on overlapping subsets of the training data. Then, the student network learns from the combined predictions of the teacher networks. In each iteration, a random weight (i.e. sampled from an exponential distribution) is assigned to each's teacher prediction. Finally, the student network's weights are averaged across multiple checkpoints after every few iterations.
3. **Label correction**: A technique to update the ground-truth label.
    a. **Gradient-based**: A technique to update the ground-truth labels using Stochastic Gradient Descent (SGD) [65], where gradients are computed with respect to the labels rather than the model parameters, as demonstrated in [59] and [66].
    b. **Cluster-based**: A technique to update the ground-truth labels by constructing clusters based on the feature space.
        - [67] proposed to compress the latent feature space using an autoencoder, with the output then used for clustering using the K-means algorithm [63]. In the subsequent supervised learning step, additional terms were introduced into the loss function to both minimize the distances between samples within the same cluster and maximize the distances between samples from different cluster. The intra-cluster loss is defined as:

$$L_{\text{intra}} = \sum_{c=1}^{K} \left(\sum_{i=1}^{N_c} \sum_{j \neq i}^{N_c} \left\| F(x_i^c; \theta) - F\left(x_j^c; \theta\right) \right\|_2 \right),$$

while the inter-cluster loss is defined as:

$$L_{\text{inter}} = -\sum_{c_1=1}^{K}\sum_{c_2\neq c_1}^{K}\left(\sum_{i=1}^{N_{c_1}}\sum_{j=1}^{Nc_2}\left\|F\left(x_i^{c_1};\theta\right)-F\left(x_j^{c_2};\theta\right)\right\|_2\right),$$

where $F(x_i^c;\theta)$ denotes the feature representation of sample x_i^c in cluster c and N_c represents the number of samples in cluster c.

c. **Label Smoothing**: A technique to soften label values from hard values (e.g. 0 or 1 in binary classification) into softer values (e.g., 0.9 and 0.1), as demonstrated in [68] to handle uncertain samples (those where the label is ambiguous or less certain).
d. **Mixup** [53] **based**: A technique to augment the training data by combining two samples, as demonstrated in [69], which studies the effect of MixMatch [70] on a histology dataset. MixMatch [70] is a technique to guess labels for unlabeled samples with low entropy and then combines labeled and pseudo-labeled samples using mixup [53].
e. **Joint Optimization** [71] proposed to update the ground-truth labels by combining the ground-truth label and the model's prediction, while introducing an additional term in the loss function to optimize the correction of noisy labels.
f. **DivideMix** [46] proposed to update the labels of clean samples by combining the ground-truth label with the average model predictions across augmentations of the presumed clean samples.

Category limitations:

a. **BadLabel Noise** [43]: The noise type referred to as BadLabel [43] generates noisy samples with a distribution indistinguishable from that of clean samples. This property may mislead the technique to update the labels of clean samples into incorrect ones.
b. **Low Noise Scenarios**: High amount of label noise is not common in medical image datasets. In datasets with only a low amount of label noise, existing label correction methods will introduce more label noise into the training data [19].

4. **Noise transition matrix estimation**: A technique to add a weight into the loss function, based on the assumption that each sample with a ground-truth label has a fixed probability of being mislabeled to an incorrect class. The weight is derived from a noise transition matrix, where the i-th row corresponds to the ground-truth class and the j-th column represents the incorrect class, as demonstrated in [72], [73], [61], [74], [75], [76], [77] and [78].

Category limitations:

a. **Assumption Validity**: The assumptions underlying these methods may not hold in real-world datasets.
b. **Complexity in Many Classes**: Noise rate estimation is challenging on datasets with many classes [33].

5. **Mean and covariance estimation methods**: A technique to model and address label noise based on the mean and covariance of data distributions.
   a. **Noise Estimation Statistics with Clusters (NESC)** [79] proposed to cluster samples based on their features using the K-means algorithm [63]. The assumption is that samples sharing the same labels will naturally group into the same cluster. Noise proportions are then estimated based on the means and covariances of the resulting clusters.
      Limitations:
      i. **Binary Classification Focus**: NESC [79] is designed for binary classification tasks.
   b. **Robust Generative classifier (RoG)** [80] assumes that the features of both clean and noisy samples follow a Gaussian distribution. In addition, [80] assumes that noisy samples are more scattered than the clean samples.
      Limitations:
      i. **Bias from Noisy Samples**: If the selected samples include noisy labels, it can introduce bias in the estimation of the mean and covariances.
      ii. **Gaussian Assumption**: The assumption that features follow a Gaussian distribution may not hold in many real-world scenarios.
6. **Robust loss function**: A technique to reduce the negative effects of noisy labels by suppressing large gradients in the loss function.
   a. **Cross Entropy Loss** [81]**,** [35], defined as
   $$L(\hat{y}, y) = -[y \ln(\hat{y}) + (1 - y) \ln(1 - \hat{y})],$$
   has a limitation that its gradients become large when the model makes incorrect predictions with high confidence (e.g. $\hat{y} \to 0$ when $y = 1$ or $\hat{y} \to 1$ when $y = 0$). This results in large updates to the model parameters, causing overfitting. To address this issue, loss functions that suppress large gradient can address this issue [3].
   b. **Generalized Cross Entropy (GCE)** [82], defined as
   $$L = (\hat{y}, y) = \frac{1-\hat{y}^q}{q},$$
   where q is a hyperparameter. Notably, as $q \to 0$, GCE simplifies to $L = \ln \hat{y}$, like the Cross Entropy Loss, which is sensitive to noisy labels. Conversely, at $q = 1$, GCE simplifies to $L = 1 - \hat{y}$, resembling the Mean Absolute Error (MAE), which is more robust to noisy labels. This property allows GCE to be applied

as a training strategy where the model starts with $q = 0$ to efficiently learn with clean samples and increase q over time. By doing so, the model learns with more samples, including potentially noisy samples, while suppressing large gradients caused by noisy samples.

c. **Local contrastive loss**: A technique to minimize the distance between two augmented views from the same sample in the latent space and maximize the distance between the augmented views of a sample and the augmented views of all other samples. The idea is to allow the model to learn from the noisy samples based on their feature representations rather than their label, thereby reducing the negative effects of noisy labels, as demonstrated in [19]. The local contrastive loss is defined as:

$$L_{\text{local}} = \sum_{i=1}^{2N} \left(-\log \frac{\exp\left(\text{sim}\left(z_i, z_j^{(i)}\right)\right)}{\sum_{k=1}^{2N} 1_{k \neq i} \times \exp(z_i, z_j)/\tau} \right),$$

where z_i denote the projection of the feature, $\cdot_j^{(i)}$ is the index of the other augmented view from the same sample,

$$\text{sim}(z_i, z_j) = \frac{z_i^T \cdot z_j}{\|z_i\|_2 + \|z_j\|_2}$$

represents the cosine similarity between z_i and z_j, τ denotes the temperature hyperparameter.

d. **Consistency loss**: A technique to encourage the model to produce similar predictions under different versions of the same input.

   i. [83] proposed to use mixture of augmentation, dropout layers and random max-pooling.

   ii. [84], [85] proposed to minimize the network predictions between a strongly augmented and weakly augmented version of the input.

   iii. [86] proposed to infer the label of weakly-augmented, unlabeled samples. The pseudo-label is only retained if the model predicts the label with high confidence. In the later stages, the model is trained to predict the strongly augmented version of the same sample.

Category limitations:

a. **Noise Type Dependency**: [82], [87], [88], [89], [90] have demonstrated robustness in training models on datasets containing noisy samples under certain assumptions [64]. However, these methods may have limitations depending on the noise types [91], such as instance-dependent noise [40], [41], [42] and BadLabel [43].

7. **Data augmentation**: A technique to generate a new sample based on the original sample.

a. [92] proposed to use weak augmentations, such as random flipping and crop-based image augmentations, for tasks involving identification of clean and noisy samples, and pseudo-labeling or inferring labels for unlabeled samples. For the backpropagation step, the study proposed to use strong augmentations, including advanced techniques such as AutoAugment [93], and transformation like rotation, inversion, and shearing to improve generalization. The approach assumes the use of weaker augmentations during the earlier epochs, followed by stronger augmentations in later stages, to avoid adversely affecting the memorization effect [92].

b. **Notable methods not focused on noisy labels**:
Augmentation policies: A technique to generate new sample based on a predefined set of rules or strategies.
    i. [93] proposed to use reinforcement learning to determine the selection and ordering of augmentation functions, with the objective of optimizing the validation loss.
    ii. [94] proposed an improvement over [93] by using grid search to identify optimal augmentation policies, which reduces the search space.

8. **Robust Optimizer**: A technique to minimize loss value.
    a. [28] proposed to identify parameters within neighborhoods characterized by uniformly low loss, rather than solely focusing on parameters with minimal loss value. The sharpness term is defined as:

$$\max_{\|\epsilon\|_2 \leq \rho} L_S(w+\epsilon) - L_S(w),$$

where ϵ denotes the perturbation, ρ is a hyperparameter, and w represents model parameters [28]. Intuitively, the expression $L_S(w+\epsilon) - L_S(w)$ measures the change in the loss due to a perturbation ϵ. The term $\max_{\|\epsilon\|_2 \leq \rho} \cdot$ finds the worst-case increase in the loss within the allowed neighborhood defined by $\|\epsilon\|_2 \leq \rho$. A large maximum increase indicates that the selected parameters is highly sensitive to perturbations, suggesting a sharp minimum. In practice, while the subtraction $L_S(w+\epsilon) - L_S(w)$ provides a conceptual understanding of sharpness, it is not strictly necessary for the optimizer. Therefore, the sharpness term can alternatively be defined as

$$\max_{\|\epsilon\|_p \leq \rho} L_S(w+\epsilon),$$

where p represents the p-norm, and $p \in [1, \infty]$. Based on this formulation, Sharpness-Aware Minimization (SAM) optimizer is defined as

$$\min_w L_S^{\mathrm{SAM}}(w) + \lambda \|w\|_2^2,$$

where $L_S^{\text{SAM}}(w) \triangleq \max_{\|\epsilon\|_p \leq \rho} L_S(w+\epsilon)$ and λ is a hyperparameter. The L_2 -regularization term, $\lambda\|w\|_2^2$, penalizes large values of w, mitigating overfitting by penalizing overly complex models.

To further simplify the optimizer, due to the constraint $\|\epsilon\|_p \leq \rho$, which ensures that ϵ is small, the optimal perturbation $e^*(w)$ can be approximated using a first-order Taylor expansion around $\epsilon = 0$, as follows

$$e^*(w) \triangleq \arg\max_{\|\epsilon\|_p \leq \rho} L_S(w+\epsilon) \approx \arg\max_{\|\epsilon\|_p \leq \rho} L_S(w) + \epsilon^T \nabla_w L_S(w),$$

where $e^*(w)$ is the optimal perturbation that maximizes the loss $L_S(w+\epsilon)$ within the allowed region $\|\epsilon\|_p \leq \rho$ and $\cdot^T$ denotes the vector transposition. Furthermore, since $L_S(w)$ is constant with respect to ϵ during the maximization process, the optimal perturbation simplifies to

$$e^*(w) = \arg\max_{\|\epsilon\|_p \leq \rho} \epsilon^T \nabla_w L_S(w).$$

Category limitations:

a. **Generalization vs. Training Loss**: Minimizing only the training loss often results in a model that performs well on the training set but poorly on the test set [28]. The training loss landscapes of deep learning models are inherently non-convex [27]. A convex landscape has a single global minimum and no local minimum, while a non-convex landscape contains both local and global minima. It is well known that sharp minima are associated with poorer generalization, as they are more sensitive to perturbations, which negatively impacts the model's ability to perform well on unseen data [27].

4 Conclusion

Our findings suggest that research in medical imaging using deep learning neglects critical factors such as optimizers, batch size, and the numbers of model parameters. Instead, the focus typically lies on the model architectures or the loss functions. This oversight may limit the potential for achieving optimal model generalization ability. Recent studies on Sharpness-Aware Minimization (SAM) optimizers [28], batch size [27], and number of model parameters [26], have showed the relationship between the training loss landscape and model generalization ability. Based on our findings, we recommend considering optimizers, balanced batch size, number of model parameters as a key strategy to ensure model generalization ability.

5 Acknowledgement

Thanks Associate Professor Huanjie Tao for his invaluable guidance.

6 References

[1] A. P. Brady, “Error and discrepancy in radiology: inevitable or avoidable?,” *Insights Imaging*, vol. 8, no. 1, pp. 171–182, Feb. 2017, doi: 10.1007/s13244-016-0534-1.

[2] H. Lu *et al.*, “Automated stent coverage analysis in intravascular OCT (IVOCT) image volumes using a support vector machine and mesh growing,” *Biomed Opt Express*, vol. 10, no. 6, p. 2809, Jun. 2019, doi: 10.1364/BOE.10.002809.

[3] Y. Wei, Y. Deng, C. Sun, M. Lin, H. Jiang, and Y. Peng, “Deep learning with noisy labels in medical prediction problems: a scoping review,” *Journal of the American Medical Informatics Association*, vol. 31, no. 7, pp. 1596–1607, Nov. 2024, doi: 10.1093/jamia/ocae108.

[4] M. Zhu, L. Zhang, L. Wang, D. Li, J. Zhang, and Z. Yi, “Robust co-teaching learning with consistency-based noisy label correction for medical image classification,” *Int J Comput Assist Radiol Surg*, vol. 18, no. 4, pp. 675–683, 2023, doi: 10.1007/s11548-022-02799-6.

[5] A. Hekler *et al.*, “Effects of Label Noise on Deep Learning-Based Skin Cancer Classification,” *Front Med (Lausanne)*, vol. 7, 2020, doi: 10.3389/fmed.2020.00177.

[6] C. Ding, T. Pereira, R. Xiao, R. J. Lee, and X. Hu, “Impact of Label Noise on the Learning Based Models for a Binary Classification of Physiological Signal,” *Sensors*, vol. 22, no. 19, 2022, doi: 10.3390/s22197166.

[7] B. Khanal, S. M. K. Hasan, B. Khanal, and C. A. Linte, “Investigating the impact of class-dependent label noise in medical image classification,” in *Medical Imaging 2023: Image Processing*, O. Colliot and I. Išgum, Eds., SPIE, 2023, p. 1246437. doi: 10.1117/12.2654420.

[8] R. K. Samala, H.-P. Chan, L. M. Hadjiiski, M. A. Helvie, and C. D. Richter, “Generalization error analysis for deep convolutional neural network with transfer learning in breast cancer diagnosis,” *Phys Med Biol*, vol. 65, no. 10, p. 105002, May 2020, doi: 10.1088/1361-6560/ab82e8.

[9] M. Büttner, L. Schneider, A. Krasowski, J. Krois, B. Feldberg, and F. Schwendicke, "Impact of Noisy Labels on Dental Deep Learning—Calculus Detection on Bitewing Radiographs," *J Clin Med*, vol. 12, no. 9, 2023, doi: 10.3390/jcm12093058.

[10] R. Jang *et al.*, "Assessment of the Robustness of Convolutional Neural Networks in Labeling Noise by Using Chest X-Ray Images From Multiple Centers," *JMIR Med Inform*, vol. 8, no. 8, 2020, doi: https://doi.org/10.2196/18089.

[11] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning From Noisy Labels With Deep Neural Networks: A Survey," *IEEE Trans Neural Netw Learn Syst*, vol. 34, no. 11, pp. 8135–8153, Nov. 2023, doi: 10.1109/TNNLS.2022.3152527.

[12] G. Algan and I. Ulusoy, "Image classification with deep learning in the presence of noisy labels: A survey," *Knowl Based Syst*, vol. 215, p. 106771, 2021, doi: https://doi.org/10.1016/j.knosys.2021.106771.

[13] X. Liang, X. Liu, and L. Yao, "Review–A Survey of Learning from Noisy Labels," *ECS Sensors Plus*, vol. 1, no. 2, p. 21401, Jun. 2022, doi: 10.1149/2754-2726/ac75f5.

[14] M. Ashraf, W. R. Q. Robles, M. Kim, Y. S. Ko, and M. Y. Yi, "A loss-based patch label denoising method for improving whole-slide image analysis using a convolutional neural network," *Sci Rep*, vol. 12, no. 1, p. 1392, Jan. 2022, doi: 10.1038/s41598-022-05001-8.

[15] A. Jaiswal, K. Ashutosh, J. F. Rousseau, Y. Peng, Z. Wang, and Y. Ding, "RoS-KD: A Robust Stochastic Knowledge Distillation Approach for Noisy Medical Imaging," in *2022 IEEE International Conference on Data Mining (ICDM)*, IEEE, Nov. 2022, pp. 981–986. doi: 10.1109/ICDM54844.2022.00118.

[16] H. Chen *et al.*, "Adaptive Cross Entropy for ultrasmall object detection in Computed Tomography with noisy labels," *Comput Biol Med*, vol. 147, p. 105763, 2022, doi: https://doi.org/10.1016/j.compbiomed.2022.105763.

[17] R. del Amor, J. Silva-Rodríguez, and V. Naranjo, "Labeling confidence for uncertainty-aware histology image classification," *Computerized Medical Imaging and Graphics*, vol. 107, p. 102231, 2023, doi: https://doi.org/10.1016/j.compmedimag.2023.102231.

[18] A. Jiménez-Sánchez *et al.*, "Curriculum learning for improved femur fracture classification: Scheduling data with prior knowledge and uncertainty," *Med Image Anal*, vol. 75, p. 102273, 2022, doi: https://doi.org/10.1016/j.media.2021.102273.

[19] C. Xue, L. Yu, P. Chen, Q. Dou, and P.-A. Heng, “Robust Medical Image Classification From Noisy Labeled Data With Global and Local Representation Guided Co-Training,” *IEEE Trans Med Imaging*, vol. 41, no. 6, pp. 1371–1382, 2022, doi: 10.1109/TMI.2021.3140140.

[20] C. Xue, Q. Dou, X. Shi, H. Chen, and P.-A. Heng, “Robust Learning at Noisy Labeled Medical Images: Applied to Skin Lesion Classification,” in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, 2019, pp. 1280–1283. doi: 10.1109/ISBI.2019.8759203.

[21] B. and G. E. and G. S. and K. M. K. and S. R. and D. S. R. and G. S. and C. D. Ghesu Florin C. and Georgescu, “Quantifying and Leveraging Classification Uncertainty for Chest Radiograph Assessment,” in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, T. and P. T. M. and S. L. H. and E. C. and Z. S. and Y. P.-T. and K. A. Shen Dinggang and Liu, Ed., Cham: Springer International Publishing, 2019, pp. 676–684.

[22] J. Irvin *et al.*, “CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 590–597, Jul. 2019, doi: 10.1609/aaai.v33i01.3301590.

[23] D. Tjandra and J. Wiens, “Leveraging an Alignment Set in Tackling Instance-Dependent Label Noise,” *Conference on Health, Inference, and Learning*, [Online]. Available: https://par.nsf.gov/biblio/10438475

[24] E. Petersen, S. Holm, M. Ganz, and A. Feragen, “The path toward equal performance in medical machine learning,” *Patterns*, vol. 4, no. 7, p. 100790, 2023, doi: https://doi.org/10.1016/j.patter.2023.100790.

[25] H. and W. J. and L. F. and D. Q. and C. G. and H. P.-A. Li Jinpeng and Cao, “Learning Robust Classifier for Imbalanced Medical Image Dataset with Noisy Labels by Minimizing Invariant Risk,” in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023*, A. and M. P. and S. S. and D. J. and S.-M. T. and T. R. Greenspan Hayit and Madabhushi, Ed., Cham: Springer Nature Switzerland, 2023, pp. 306–316.

[26] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning requires rethinking generalization,” in *International Conference on Learning Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=Sy8gdB9xx

[27] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, “On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima,” in *International Conference on Learning Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=H1oyRlYgg

[28] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, “Sharpness-aware Minimization for Efficiently Improving Generalization,” in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=6Tm1mposlrM

[29] W. Li *et al.*, “PathAL: An Active Learning Framework for Histopathology Image Analysis,” *IEEE Trans Med Imaging*, vol. 41, no. 5, pp. 1176–1187, 2022, doi: 10.1109/TMI.2021.3135002.

[30] A. Blum and T. Mitchell, “Combining labeled and unlabeled data with co-training,” in *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, in COLT’ 98. New York, NY, USA: Association for Computing Machinery, 1998, pp. 92–100. doi: 10.1145/279943.279962.

[31] Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin, “A co-regularization approach to semi-supervised learning with multiple views,” *Proceedings of ICML Workshop on Learning With Multiple Views*, pp. 74–79, Aug. 2005.

[32] B. Han *et al.*, “Co-teaching: robust training of deep neural networks with extremely noisy labels,” in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, in NIPS’18. Red Hook, NY, USA: Curran Associates Inc., 2018, pp. 8536–8546.

[33] H. Wei, L. Feng, X. Chen, and B. An, “Combating Noisy Labels by Agreement: A Joint Training Method with Co-Regularization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020.

[34] H. Jeffreys, *Theory of Probability*, 2nd ed. Oxford University Press, 1948.

[35] S. Kullback and R. A. Leibler, “On Information and Sufficiency,” *The Annals of Mathematical Statistics*, vol. 22, no. 1, p. 80, 1951, doi: 10.1214/aoms/1177729694.

[36] S. Kullback, *Information Theory and Statistics*. John Wiley & Sons, 1959.

[37] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari, “Learning with Noisy Labels,” in *Advances in Neural Information Processing Systems*, C. J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2013. [Online]. Available:

https://proceedings.neurips.cc/paper_files/paper/2013/file/3871bd64012152bfb53fdf04b401193f-Paper.pdf

[38] D. Angluin and P. Laird, "Learning From Noisy Examples," *Mach Learn*, vol. 2, no. 4, pp. 343–370, 1988, doi: 10.1023/A:1022873112823/METRICS.

[39] B. van Rooyen, A. Menon, and R. C. Williamson, "Learning with Symmetric Label Noise: The Importance of Being Unhinged," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/file/45c48cce2e2d7fbdea1afc51c7c6ad26-Paper.pdf

[40] X. Xia *et al.*, "Part-dependent Label Noise: Towards Instance-dependent Label Noise," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., Curran Associates, Inc., 2020, pp. 7597–7610. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/5607fe8879e4fd269e88387e8cb30b7e-Paper.pdf

[41] Y. Zhang, S. Zheng, P. Wu, M. Goswami, and C. Chen, "Learning with Feature-Dependent Label Noise: A Progressive Approach," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=ZPa2SyGcbwh

[42] P. Chen, J. Ye, G. Chen, J. Zhao, and P. A. Heng, "Beyond Class-Conditional Assumption: A Primary Attempt to Combat Instance-Dependent Label Noise," *35th AAAI Conference on Artificial Intelligence, AAAI 2021*, vol. 13A, pp. 11442–11450, 2021, doi: 10.1609/AAAI.V35I13.17363.

[43] J. Zhang *et al.*, "BadLabel: A Robust Perspective on Evaluating and Enhancing Label-Noise Learning," *IEEE Trans Pattern Anal Mach Intell*, vol. 46, no. 6, pp. 4398–4409, Jun. 2024, doi: 10.1109/TPAMI.2024.3355425.

[44] X. Yu, B. Han, J. Yao, G. Niu, I. Tsang, and M. Sugiyama, "How does Disagreement Help Generalization against Label Corruption?," in *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri and R. Salakhutdinov, Eds., in Proceedings of Machine Learning Research, vol. 97. PMLR, Oct. 2019, pp. 7164–7173. [Online]. Available: https://proceedings.mlr.press/v97/yu19b.html

[45] L. Ju *et al.*, “Improving Medical Images Classification With Label Noise Using Dual-Uncertainty Estimation,” *IEEE Trans Med Imaging*, vol. 41, no. 6, pp. 1533–1546, 2022, doi: 10.1109/TMI.2022.3141425.

[46] J. Li, R. Socher, and S. C. H. Hoi, “DivideMix: Learning with Noisy Labels as Semi-supervised Learning,” in *International Conference on Learning Representations*, 2020.

[47] H. Permuter, J. Francos, and I. Jermyn, “A study of Gaussian mixture models of color and texture features for image classification and segmentation,” *Pattern Recognit*, vol. 39, no. 4, pp. 695–706, 2006, doi: https://doi.org/10.1016/j.patcog.2005.10.028.

[48] S. J. Roberts, D. Husmeier, W. Penny, and lead Rezek, “Bayesian Approaches to Gaussian Mixture Modeling,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1133–1142, Nov. 1998, doi: 10.1109/34.730550.

[49] M. Raghu *et al.*, “Direct Uncertainty Prediction for Medical Second Opinions,” in *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri and R. Salakhutdinov, Eds., in Proceedings of Machine Learning Research, vol. 97. PMLR, Nov. 2019, pp. 5281–5290. [Online]. Available: https://proceedings.mlr.press/v97/raghu19a.html

[50] Y. Gal and Z. Ghahramani, “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning,” in *Proceedings of The 33rd International Conference on Machine Learning*, M. F. Balcan and K. Q. Weinberger, Eds., in Proceedings of Machine Learning Research, vol. 48. New York, New York, USA: PMLR, Nov. 2016, pp. 1050–1059. [Online]. Available: https://proceedings.mlr.press/v48/gal16.html

[51] J. Cohen, “A Coefficient of Agreement for Nominal Scales,” *Educ Psychol Meas*, vol. 20, no. 1, pp. 37–46, Apr. 1960, doi: 10.1177/001316446002000104.

[52] K. A. Hallgren, “Computing Inter-Rater Reliability for Observational Data: An Overview and Tutorial,” *Tutor Quant Methods Psychol*, vol. 8, no. 1, pp. 23–34, 2012, doi: 10.20982/tqmp.08.1.p023.

[53] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond Empirical Risk Minimization,” in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=r1Ddp1-Rb

[54] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, “LoOP: local outlier probabilities,” in *Proceedings of the 18th ACM Conference on Information and Knowledge*

Management, in CIKM '09. New York, NY, USA: Association for Computing Machinery, 2009, pp. 1649–1652. doi: 10.1145/1645953.1646195.

[55] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning*, in ICML'20. JMLR.org, 2020.

[56] P. Chen, B. Ben Liao, G. Chen, and S. Zhang, "Understanding and Utilizing Deep Neural Networks Trained with Noisy Labels," in *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri and R. Salakhutdinov, Eds., in Proceedings of Machine Learning Research, vol. 97. PMLR, Nov. 2019, pp. 1062–1070. [Online]. Available: https://proceedings.mlr.press/v97/chen19g.html

[57] W. Feng *et al.*, "Graph random neural networks for semi-supervised learning on graphs," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, in NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[58] J. Xiang *et al.*, "Automatic diagnosis and grading of Prostate Cancer with weakly supervised learning on whole slide images," *Comput Biol Med*, vol. 152, p. 106340, 2023, doi: https://doi.org/10.1016/j.compbiomed.2022.106340.

[59] J. Liu, R. Li, and C. Sun, "Co-Correcting: Noise-Tolerant Medical Image Classification via Mutual Label Correction," *IEEE Trans Med Imaging*, vol. 40, no. 12, pp. 3580–3592, 2021, doi: 10.1109/TMI.2021.3091178.

[60] J. Ren *et al.*, "OCRFinder: a noise-tolerance machine learning method for accurately estimating open chromatin regions," *Front Genet*, vol. 14, 2023, doi: 10.3389/fgene.2023.1184744.

[61] T. Liu and D. Tao, "Classification with Noisy Labels by Importance Reweighting," *IEEE Trans Pattern Anal Mach Intell*, vol. 38, no. 3, pp. 447–461, 2016, doi: 10.1109/TPAMI.2015.2456899.

[62] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008, [Online]. Available: http://jmlr.org/papers/v9/vandermaaten08a.html

[63] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans Inf Theory*, vol. 28, no. 2, pp. 129–137, 1982, doi: 10.1109/TIT.1982.1056489.

[64] W. Luo *et al.*, "Estimating Per-Class Statistics for Label Noise Learning," *IEEE Trans Pattern Anal Mach Intell*, 2024, doi: 10.1109/TPAMI.2024.3466182.

[65] H. Robbins and S. Monro, “A Stochastic Approximation Method,” *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400 – 407, 1951, doi: 10.1214/aoms/1177729586.

[66] P. Shi, J. Xin, and N. Zheng, “Correcting Pseudo Labels with Label Distribution for Unsupervised Domain Adaptive Vulnerable Plaque Detection,” in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 2021, pp. 3225–3228. doi: 10.1109/EMBC46164.2021.9629833.

[67] C. Ding *et al.*, “Learning From Alarms: A Robust Learning Approach for Accurate Photoplethysmography-Based Atrial Fibrillation Detection Using Eight Million Samples Labeled With Imprecise Arrhythmia Alarms,” *IEEE J Biomed Health Inform*, vol. 28, no. 5, pp. 2650–2661, 2024, doi: 10.1109/JBHI.2024.3360952.

[68] H. H. Pham, T. T. Le, D. Q. Tran, D. T. Ngo, and H. Q. Nguyen, “Interpreting chest X-rays via CNNs that exploit hierarchical disease dependencies and uncertainty labels,” *Neurocomputing*, vol. 437, pp. 186–194, 2021, doi: https://doi.org/10.1016/j.neucom.2020.03.127.

[69] J. V. Pulido *et al.*, “Semi-Supervised Classification of Noisy, Gigapixel Histology Images,” in *2020 IEEE 20th International Conference on Bioinformatics and Bioengineering (BIBE)*, 2020, pp. 563–568. doi: 10.1109/BIBE50027.2020.00097.

[70] D. Berthelot, N. Carlini, I. Goodfellow, A. Oliver, N. Papernot, and C. Raffel, “MixMatch: a holistic approach to semi-supervised learning,” in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA: Curran Associates Inc., 2019.

[71] D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa, “Joint Optimization Framework for Learning with Noisy Labels,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5552–5560. doi: 10.1109/CVPR.2018.00582.

[72] D. Cheng *et al.*, “Class-Dependent Label-Noise Learning with Cycle-Consistency Regularization,” in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., Curran Associates, Inc., 2022, pp. 11104–11116. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/47f75e809409709c6d226ab5ca0c9703-Paper-Conference.pdf

[73] X. Li, T. Liu, B. Han, G. Niu, and M. Sugiyama, “Provably End-to-end Label-noise Learning without Anchor Points,” in *Proceedings of the 38th International Conference on Machine Learning*, M. Meila and T. Zhang, Eds., in Proceedings of Machine

Learning Research, vol. 139. PMLR, Oct. 2021, pp. 6403–6413. [Online]. Available: https://proceedings.mlr.press/v139/li21l.html

[74] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu, "Making Deep Neural Networks Robust to Label Noise: A Loss Correction Approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017.

[75] X. Xia *et al.*, "Are Anchor Points Really Indispensable in Label-Noise Learning?," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, Eds., Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/9308b0d6e5898366a4a986bc33f3d3e7-Paper.pdf

[76] Y. Yao *et al.*, "Dual T: Reducing Estimation Error for Transition Matrix in Label-noise Learning," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., Curran Associates, Inc., 2020, pp. 7260–7271. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/512c5cad6c37edb98ae91c8a76c3a291-Paper.pdf

[77] L. I. N. Yong *et al.*, "A Holistic View of Label Noise Transition Matrix in Deep Learning and Beyond," in *The Eleventh International Conference on Learning Representations* , 2023. [Online]. Available: https://openreview.net/forum?id=aFzaXRImWE

[78] Y. Zhang, G. Niu, and M. Sugiyama, "Learning Noise Transition Matrix from Only Noisy Labels via Total Variation Regularization," in *Proceedings of the 38th International Conference on Machine Learning*, M. Meila and T. Zhang, Eds., in Proceedings of Machine Learning Research, vol. 139. PMLR, Oct. 2021, pp. 12501–12512. [Online]. Available: https://proceedings.mlr.press/v139/zhang21n.html

[79] W. Gao, T. Zhang, B.-B. Yang, and Z.-H. Zhou, "On the noise estimation statistics," *Artif Intell*, vol. 293, p. 103451, 2021, doi: https://doi.org/10.1016/j.artint.2021.103451.

[80] K. Lee, S. Yun, K. Lee, H. Lee, B. Li, and J. Shin, "Robust Inference via Generative Classifiers for Handling Noisy Labels," in *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri and R. Salakhutdinov, Eds., in Proceedings of Machine Learning Research, vol. 97. PMLR, Oct. 2019, pp. 3763–3772. [Online]. Available: https://proceedings.mlr.press/v97/lee19f.html

[81] C. E. Shannon, “A mathematical theory of communication,” *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948, doi: 10.1002/j.1538-7305.1948.tb01338.x.

[82] Z. Zhang and M. R. Sabuncu, “Generalized cross entropy loss for training deep neural networks with noisy labels,” in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, in NIPS’18. Red Hook, NY, USA: Curran Associates Inc., 2018, pp. 8792–8802.

[83] M. Sajjadi, M. Javanmardi, and T. Tasdizen, “Regularization with stochastic transformations and perturbations for deep semi-supervised learning,” in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, in NIPS’16. Red Hook, NY, USA: Curran Associates Inc., 2016, pp. 1171–1179.

[84] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V Le, “Unsupervised data augmentation for consistency training,” in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, in NIPS ’20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[85] D. Berthelot *et al.*, “ReMixMatch: Semi-Supervised Learning with Distribution Matching and Augmentation Anchoring,” in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=HklkeR4KPB

[86] K. Sohn *et al.*, “FixMatch: simplifying semi-supervised learning with consistency and confidence,” in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, in NIPS ’20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[87] X. Ma, H. Huang, Y. Wang, S. Romano, S. Erfani, and J. Bailey, “Normalized Loss Functions for Deep Learning with Noisy Labels,” in *Proceedings of the 37th International Conference on Machine Learning*, H. D. III and A. Singh, Eds., in Proceedings of Machine Learning Research, vol. 119. PMLR, Oct. 2020, pp. 6543–6553. [Online]. Available: https://proceedings.mlr.press/v119/ma20c.html

[88] L. Feng, S. Shu, Z. Lin, F. Lv, L. Li, and B. An, “Can Cross Entropy Loss Be Robust to Label Noise?,” in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, C. Bessiere, Ed., International Joint Conferences on Artificial Intelligence Organization, Oct. 2020, pp. 2206–2212. doi: 10.24963/ijcai.2020/305.

[89] Y. Liu and H. Guo, “Peer loss functions: learning from noisy labels without knowing noise rates,” in *Proceedings of the 37th International Conference on Machine Learning*, in ICML’20. JMLR.org, 2020.

[90] X. Xia *et al.*, “Regularly Truncated M-Estimators for Learning With Noisy Labels,” *IEEE Trans Pattern Anal Mach Intell*, vol. 46, no. 5, pp. 3522–3536, 2024, doi: 10.1109/TPAMI.2023.3347850.

[91] F. R. Cordeiro and G. Carneiro, “A Survey on Deep Learning with Noisy Labels: How to train your model when you cannot trust on the annotations?,” in *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2020, pp. 9–16. doi: 10.1109/SIBGRAPI51738.2020.00010.

[92] K. Nishi, Y. Ding, A. Rich, and T. Hollerer, “Augmentation Strategies for Learning With Noisy Labels,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 8022–8031.

[93] E. D. Cubuk, B. Zoph, D. Mané, V. Vasudevan, and Q. V Le, “AutoAugment: Learning Augmentation Strategies From Data,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 113–123. doi: 10.1109/CVPR.2019.00020.

[94] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V Le, “Randaugment: Practical automated data augmentation with a reduced search space,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 3008–3017. doi: 10.1109/CVPRW50498.2020.00359.

7 Appendix A

This section provides a concise reference describing the term used throughout this paper.

Table 7.1 Key Terms

Term	Description
Label noise/uncertainty	Data with incorrect labels, such as mislabeling or observer variability
Observer variability	A sample with multiple different labels due to expert disagreement

8 Appendix B

This section provides a concise how to of the handling methods discussed throughout this paper.

Table 8.1 Co-teaching

Co-teaching [32]
Input: Samples with clean and noisy labels Process: Select clean samples Each network select clean samples (i.e. with lowest loss) for its peer network $$R(T) = 1 - \min\left\{\frac{T}{T_k}\tau, \tau\right\}$$ $R(T)$ is the selection percentage of these clean samples over the entire dataset, where T is the current epoch, T_k is the predefined threshold epoch at which $R(T)$ reaches its minimum value, τ is the noise rate $R(T)$ decreases over time to focus more on clean samples as training progress (i.e. to prevent overfitting to noisy labels)

Table 8.2 Calculate Disagreement Noise

Calculate Disagreement Noise by calculating variance in annotator counts
Input: Sample x_i: an image that is labelled by multiple annotators Suppose we have annotations from 3 doctors $Y_i = [0,0,1]$ Process Calculate the probability of each class

$$P_i = \left[\frac{2}{3}, \frac{1}{3}\right]$$

Calculate the Uncertainty of Disagreement (UoD)

$$UoD_i = 1 - \sum_{j=1}^{k} \left(p_i^j\right)^2$$

where i is the sample index, k is the number of classes and p is the probability of a class

$$UoD_i = 1 - \left(\left(\frac{2}{3}\right)^2 + \left(\frac{1}{3}\right)^2\right) \approx 0.44$$

Calculate the improved Direct Uncertainty Prediction (iUoD)
by adjusting the UoD with the number of annotators

$$iUoD_i = \left[\min\left(\sum_j 1_{y_i^j = g_c}\right)\right]^{\eta} \times UoD_i$$
$$= [\min(\text{count of Class 0 , ..., count of Class k})]^{\eta} \times UoD_i$$

where j is the annotator index, g_c is the specific class label in the set of possible classes (i.e. g_0 represents Class 0; e.g. $g_0 = 2,\ g_1 = 1$), η is a hyper-parameter
Let say $\eta = 1$

$$iUoD_i = (1)^1 \times 0.44 = 0.44$$

If $iUoD_i = 0$, then there is no disagreement noise for that image
Filter out samples with a high score of $\boldsymbol{UoD_i}$
If $UoD_i > t_{UoD}$, then filter out the sample.
where t_{UoD} is a threshold

Table 8.3 Calculate single-target noise

Calculate single-target noise using Monte-Carlo Dropout (MC-Dropout)

Input: Classes: Class 0, Class 1, and Class 2
Sample x_i
Process
Insert dropout layers in the neural network
Calculate the model's confidence in each class
Let's say, 5 forward passes ($T = 5$), we run the model on the sample x_i
$p_i^{(1)} = [0.60, 0.30, 0.10]$, $p_i^{(2)} = [0.50, 0.40, 0.10]$, $p_i^{(3)} = [0.55, 0.35, 0.10]$
Calculate the mean predictive probability

$$m_i = \frac{1}{T}\sum_{t=1}^{T} p_i^{(t)}$$

$$m_i = \frac{1}{3}([0.60,0.30,0.10] + [0.50,0.40,0.10] + [0.55,0.35,0.10])$$
$$= [0.55,0.35,0.10]$$

Calculate the uncertainty of a single-target label

$$UoSL_i = -\sum_{j=1}^{k} m_i^j \times \log\left(m_i^j\right)$$

where $m_i = \frac{1}{T}\sum_{t=1}^{T} p_i^t$ and j corresponds to the j-th class

$$UoSL_i = -(0.55 \log 0.55 + 0.35 \log 0.35 + 0.10 \log 0.10) \approx 0{,}40$$

If $UoSL_i(x_i) = 0$ then there is no single-target noise

Calculate the weights from $\boldsymbol{UoSL_i}$

$$w_i = \begin{cases} 1 - nUoSL_i, & nUoSL_i > t_{UoSL} \\ 1, & nUoSL_i \leq t_{UoSL} \end{cases}$$

where the $nUoSL_i$ is the normalized $UoSL$ mapped to $[0,1]$ and t_{UoSL} is a threshold value

To normalize, $nUoSL_i = \frac{UoSL_i - \min(UoSL)}{\max(UoSL) - \min(UoSL)}$

Let's say $t_{UoSL} = 0.6$

Suppose $UoSL_0 = 0.2, UoSL_1 = 0.5, UoSL_2 = 0.7$

So, $nUoSL_0 = \frac{0.2-0.2}{0.7-0.2} = 0, nUoSL_1 = \frac{0.5-0.2}{0.7-0.2} = 0.5, nUoSL_2 = \frac{0.7-0.2}{0.7-0.2} = 1$

Therefore, $w_0 = 1, w_1 = 1, w_2 = 1 - 1 = 0$

Uncertain samples (i.e. high $UoSL$ scores) received reduced weights, decreasing their influence on the training

Apply the weights

$$L_{wCE}(X) = -\sum_{i=1}^{z} w_i \times (q_i \times \log p_i + (1 - q_i) \times \log(1 - p_i))$$

where p_i denotes the predictions and q_i denotes the ground-truths

This equation is for binary classification. Meanwhile, Equation (1) and the Experiments section show a multi-class classification task. So, we will use the equation from PyTorch with assumption that the reduction = 'sum'

$$\ell(x, y) = \sum_{n=1}^{N} l_n$$

$$l_n = -w_{y_n} \ln \frac{\exp\left(x_{n,y_n}\right)}{\sum_{c=1}^{C} \exp\left(x_{n,c}\right)}$$

Suppose $q_i = 1, p_i = [0.7, 0.2, 0.1]$

$$l_0 = -1 \ln \frac{e^{0.2}}{e^{0.7} + e^{0.2} + e^{0.1}} \approx 1.26$$

Table 8.4 Boosting-based Curriculum Training

Boosting-based Curriculum Training
Process Train the model with less noisy samples (i.e. $\boldsymbol{UoD_i < t_{UoD}}$) samples. Reweight the filtered samples with $\boldsymbol{UoSL_i}$ Note: reweighting at each training iteration

Table 8.5 Joint Training with Co-Regularization (JoCoR)

Joint Training with Co-Regularization (JoCoR) [33]

Input: true label $y_0 = [1,0,0]$, $y_1 = [0,1,0]$, the network predictions $p_1(x_0) = [0.7,0.2,0.1]$, $p_2(x_0) = [0.6,0.3,0.1]$, $p_1(x_1) = [0.1,0.8,0.1]$, $p_2(x_1) = [0.2,0.7,0.1]$

Process

Calculate the Cross Entropy Loss

$$\ell_{\text{sup}} = \ell_{\text{C1}}(x_i, y_i) + \ell_{\text{C2}}(x_i, y_i)$$

$$= -\sum_{i=1}^{N}\sum_{m=1}^{M} y_i \log\left(p_1^m(x_i)\right) - \sum_{m=1}^{M} y_i \log\left(p_2^m(x_i)\right)$$

$$\ell_{\text{C1}}(x_i, y_i) = -\left((1 \times \log 0.7 + 0 + 0) + (0 + 1 \times \log 0.8 + 0)\right)$$

$$\approx 0.5798$$

$$\ell_{\text{C2}}(x_i, y_i) = -\left((1 \times \log(0.6) + 0 + 0) + (0 + 1 \times \log 0.7 + 0)\right)$$

$$\approx 0.8675$$

$$\ell_{\text{sup}} \approx (0.5798 + 0.8675) \approx 1.4473$$

ℓ_{sup} is to penalize disagreement between networks and amplify the penalty

Calculate Contrastive Loss

$$l_{\text{con}} = D_{\text{KL}}(p_1||p_2) + D_{\text{KL}}(p_2||p_1)$$

where KL refers to Kullback-Leibler (KL) Divergence

$$D_{\text{KL}}(p_1||p_2) = \sum_{i=1}^{N}\sum_{m=1}^{M} p_1^m(x_i) \log\frac{p_1^m(x_i)}{p_2^m(x_i)}$$

where m is the class, p_i refers to the networks' predictions

$$D_{\text{KL}}(p_2||p_1) = \sum_{i=1}^{N}\sum_{m=1}^{M} p_2^m(x_i) \log\frac{p_2^m(x_i)}{p_1^m(x_i)}$$

$$D_{KL}(p_1||p_2) \approx \left(0.7 \log\frac{0.7}{0.6} + 0.2 \log\frac{0.2}{0.3} + 0\right) + \left(0.1 \log\frac{0.1}{0.2} + 0.8 \log\frac{0.8}{0.7} + 0\right)$$

$$\approx 0.0644$$

$$D_{KL}(p_2||p_1) = \left(0.6 \log\frac{0.6}{0.7} + 0.3 \log\frac{0.3}{0.2} + 0\right) + \left(0.2 \log\frac{0.2}{0.1} + 0.7 \log\frac{0.7}{0.8} + 0\right)$$

$$\approx 0.0291$$

$$l_{con} \approx 0{,}0268 + 0{,}0291 = 0{,}0935$$

If the prediction probability between networks match, then $D_{KL} = 0$

Calculate the Joint Loss

$$\ell(x_i) = (1 - \lambda) \times \ell_{\text{sup}}(x_i, y_i) + \lambda \times \ell_{\text{con}}(x_i)$$

where l_{con} is the contrastive loss, λ is the co-regularization hyper-parameter (i.e. $\lambda = 0$ means joint training without co-regularization)

Let's say $\lambda = 0.05$ (note: in the paper, in the experiments section, $\lambda = [0.00,0.05,0.10, \ldots ,0.95]$)

$$\ell(x_i) = (1 - 0.05) \times 1.4473 + 0.05 \times 0.0935 \approx 1.37961$$

Select Clean Samples (i.e. samples with smallest loss)

Same as Co-teaching [A18]: The selection percentage of these clean samples over the entire dataset decreases overtime to avoid overfitting to noise

Suppose $\ell(x_0) \approx 0.4620$, $\ell(x_1) \approx 0.3180$

$$R(t) = 1 - \min\left\{\frac{t}{T_k}\tau, \tau\right\}$$

where t is the current epoch, T_k is the predefined threshold epoch at which $R(t)$ reaches its minimum value, τ is the noise rate

Let's say $t = 5, T_k = 10, \tau = 0.2$

$$R(t) = 1 - \min\left\{\frac{5}{10} \times 0.2, 0.2\right\} = 1 - 0.1 = 0.9$$

Table 8.6 Co-Divide

Co-Divide
Input: Loss values Using a Gaussian Mixture Model to separate clean and noisy samples $\mathcal{W}^{(2)} = \text{GMM}(\mathcal{X}, \mathcal{Y}, \theta^{(1)})$ $\theta^{(1)}$ outputs the loss of each sample. GMM outputs the clean probability of each sample. $\mathcal{W}^{(1)} = \text{GMM}(\mathcal{X}, \mathcal{Y}, \theta^{(2)})$ Notice: $\theta^{(2)}$ instead of $\theta^{(1)}$ outputs the losses for $\mathcal{W}^{(1)}$. The idea is to prevent either network from overfitting to its own errors $\mathcal{W}^{(k)}$ is the clean probabilities for $\theta^{(k)}$, k is the network number, $k \in \{1,2\}$, $\mathcal{W}^{(k)}$ is used to assigns clean samples to a labelled set and noisy samples to an unlabelled set $\mathcal{X}_e^{(k)} = \{(x_i, y_i, w_i) \mid w_i \geq \tau, \forall (x_i, y_i, w_i) \in (\mathcal{X}, \mathcal{Y}, \mathcal{W}^{(k)})\}$ where $\mathcal{X}_e^{(k)}$ is the labelled set for $\theta^{(k)}$, τ is the clean probability threshold $\mathcal{U}_e^{(k)} = \{x_i \mid w_i < \tau, \forall (x_i, w_i) \in (\mathcal{X}, \mathcal{W}^{(k)})\}$ where $\mathcal{U}_e^{(k)}$ is the unlabelled set for $\theta^{(k)}$

Table 8.7 MixMatch with Label Co-Refinement and Co-Guessing

MixMatch [70] with Label Co-Refinement and Co-Guessing
Process Augment $\hat{x}_{b,m} = \text{Augment}(x_b)$ $\hat{u}_{b,m} = \text{Augment}(u_b)$ where $\hat{x}_{b,m}$ is the clean sample augmented at indices b and m $\hat{u}_{b,m}$ is the noisy sample augmented at indices b and m B is the mini-batch size, M is the number of augmentations,

Augment function uses the MixUp augmentation (i.e. encouraging the model to have linear behaviour between samples) (according to [46])
x_b is a clean sample at index b
Label Co-Refinement

$$p_b = \frac{1}{M}\sum_m \mathrm{p}_{\mathrm{model}}\left(\hat{x}_{b,m}; \theta^{(k)}\right)$$

where
p_b is the average the predictions across augmentations of x_b
$\mathrm{p}_{\mathrm{model}}$ represents the network's predicted class probabilities
$\hat{x}_{b,m}$ is a clean sample augmented at indices b and m

$$\bar{y}_b = w_b y_b + (1 - w_b) p_b$$

$\bar{y}_b$ is the refined ground-truth label guided by the clean probability produced by the other network
Temperature Sharpening

$$\hat{y}_b = \mathrm{Sharpen}(\bar{y}_b, T) = \bar{y}_b^{c\frac{1}{T}} \Big/ \sum_{c=1}^{C} \bar{y}_b^{c\frac{1}{T}}, \text{ for } c = 1, 2, \dots, C$$

where T is the sharpening temperature (a hyper-parameter)
$\hat{y}_b$ is the sharpened and refined label
Label Co-Guessing

$$\bar{q}_b = \frac{1}{2M}\sum_m \left(\mathrm{p}_{\mathrm{model}}\left(\hat{u}_{b,m}; \theta^{(1)}\right) + \mathrm{p}_{\mathrm{model}}\left(\hat{u}_{b,m}; \theta^{(2)}\right)\right)$$

where $\bar{q}_b$ is the average predictions from both networks across augmentations of u_b
$\hat{u}_b$ is the noisy sample augmented at index b and m
Temperature Sharpening

$$q_b = \mathrm{Sharpen}(\bar{q}_b, T) = \bar{q}_b^{c\frac{1}{T}} \Big/ \sum_{c=1}^{C} \bar{q}_b^{c\frac{1}{T}}, \text{ for } c = 1,2, \dots, C$$

where q_b is the sharpened and refined label

Table 8.8 Robust DivideMix

Robust DivideMix [43]
Input: a pair of DNNs parameterized by θ_1, θ_2, noisy set $D' = (X, Y')$, selection threshold T_p and T_c, MixMatch epochs E
Output: A pair of optimized DNNs parameterized by θ_1^E, θ_2^E for making predictions jointly
Process
// Stage I: Initialization of the pair of DNNs $\boldsymbol{\theta_1}, \boldsymbol{\theta_2}$
$\theta_1^0, \theta_2^0 = \mathrm{WarmUp}(D', \theta_1, \theta_2)$ // Use D' to conduct standard training for a few epochs
Uses Confidence Penalty (CP):
By making per-sample loss distributions more distinguishable.

A negative entropy term ($\mathcal{H}$) is added to the loss function Eq. (1) to penalize overconfident predictions and increase the per-sample losses.
The negative entropy term is only used in the warm-up phase

$$\mathcal{H}\big(f_\theta(x)\big) = -\frac{1}{N} f_\theta(x) \log\big(f_\theta(x)\big)$$

Eq. (1) $\underset{f\in\mathcal{F}}{\arg\min} \frac{1}{n}\sum_{i=1}^{n} \ell\big(y_i, f(x_i)\big)$
where f denotes a classifier (i.e., a DNN in this paper), $\mathcal{F}$ is the hypothesis space, ℓ is the loss function for optimization

$$\ell\big(y_i, f(x_i)\big) = L = \{l_1, \dots, l_N\}^T$$

$$l_n = -\log \frac{\exp(x_n, y_n)}{\sum_{c=1}^{C} \exp\,(x_n, c)}$$

where x is the input, y is the target, C is the number of classes, N is the number of samples

$$\ell_{\text{regularilized}} = \ell\big(y_i, f(x_i)\big) + \mathcal{H}\big(f_\theta(x_i)\big)$$

Perturb $\boldsymbol{y_i'}$ by Eq. (8)

$$\widetilde{D} = \{(x_i, \tilde{y}_1)\}_{i=1}^{n}$$

$\widetilde{D}$ refers to the entire dataset where the labels y_i' (original noisy labels) have been perturbed to $\tilde{y}_i$ for each data point x_i

$$\tilde{Y} = Y' + \lambda \nabla_{Y'} \ell\big(Y', f(X)\big)$$

where $Y' \in \mathbb{R}^{n\times C}$ is a one hot form of noisy labels, $\tilde{Y} \in \mathbb{R}^{n\times C}$ is an adversarially perturbed variant of Y', λ is the step size, $\nabla_{Y'} \ell\big(Y', f(X)\big)$ is the gradient of the loss function with respect to these noisy labels, indicating how to adjust them to minimize the loss
Calculate the posterior probability of each sample

$$W_1^p = \text{BayesGMM}\big(\widetilde{D}, \theta_2^0\big), W_2^p = \text{BayesGMM}\big(\widetilde{D}, \theta_1^0\big)$$

"Two students correcting each other's homework"
W^p refers to posterior probability (hence the "p") that the i-th data point is clean, whose element w_i is the posterior probability that the $\boldsymbol{i}$-th data point has a smaller loss value, indicating that it is likely to belong to the "clean" label set
$\text{BayesGMM}(\cdot)$ to model per-sample loss distributions.
When the loss distribution of BadLabel is unimodal rather than bimodal, the convergence speed of BayesGMM is slow when the number of components is preset to two. The convergence speed enables us to determine whether the clean and noisy labels can be effectively differentiated during the selection of clean labels
These loss distributions are expected to be bimodal, meaning there should be two distinct clusters: one for clean labels and one for noisy labels
for $\boldsymbol{k} = \mathbf{1}, \mathbf{2}$ do
k is the network i.e. θ_0 and θ_1
$\mathcal{X}_k = \big\{(x_i, y_i') | w_i^p \geq \mathrm{T}_p, \forall\big(x_i, y_i', w_i^p\big) \in \big(X, Y', W_k^p\big)\big\}$ // (mostly) clean labeled set
$\mathcal{X}_k$ is the set of samples with posterior probabilities above the threshold T_p
$\mathcal{U}_k = \{x_i | (x_i, y_i') \in D' \wedge (x_i, y_i') \notin \mathcal{X}_k\}$ // unlabeled set
$\wedge$ is a logical AND operator

$(x_i, y_i') \in D'$ means the sample (x_i, y_i') is part of the entire noisy dataset D', where x_i is the input and y_i' is the noisy label

$(x_i, y_i') \notin \mathcal{X}_k$ means that the sample (x_i, y_i') is not included in the mostly clean labeled set $\mathcal{X}_k$ for the network k

end for

Obtain $\boldsymbol{\theta_1^1}$ and $\boldsymbol{\theta_2^1}$ by Eq. (10) for a single epoch (i.e., $\boldsymbol{e = 0 \rightarrow e = 1}$)

$\theta_k^{e+1} = \text{MixMatch}(\mathcal{X}_k, \mathcal{U}_k, \theta_k^e)$, with $k \in \{1,2\}$

where $e \in \{1,2, \dots, E\}$ is the index of the total E training epochs,

// Stage II: Pair-wise training of $\boldsymbol{\theta_1^1}$ and $\boldsymbol{\theta_2^1}$ by Eq. (10) for a single epoch

$$W_i^c = W_2^p, W_2^c = W_1^p$$

W^c represent the current or updated posterior probabilities during the training process

The current posteriors W_1^c are initialized with the posterior probabilities W_2^p, which were calculated using θ_1 during the warm-up phase

for epoch $e = 1, \dots, E$ do

if $\text{BayesGMM}(D', \theta_2^e)$ is converged then

$$W_1^c = \text{BayesGMM}(D', \theta_2^e)$$

If θ_2^e is good enough in separating clean and noisy labels, then update W_1^c

end if

if $\text{BayesGMM}(D', \theta_1^e)$ is converged then

$$W_2^c = \text{BayesGMM}(D', \theta_1^e)$$

end if

for $k = 1,2$ do

$\mathcal{X}_k = \{(x_i, y_i') | w_i^c \geq \text{T}_c, \forall (x_i, y_i^c, w_i^c) \in (X, Y', W_k^c)\}$ // (mostly) clean labeled set

$\mathcal{U}_k = \{x_i | (x_i, y_i') \in D' \wedge (x_i, y_i') \notin \mathcal{X}_k\}$ // unlabeled set

end for

Obtain $\boldsymbol{\theta_1^{e+1}}$ and $\boldsymbol{\theta_2^{e+1}}$ by Eq. (10) (i.e., $\boldsymbol{e \rightarrow e + 1}$).

In other words, at each epoch, $\boldsymbol{W_k^c}$, $\boldsymbol{\mathcal{X}_k}$, $\boldsymbol{\theta_k^{e+1}}$ may update;

W_k^c is calculated with $W_k^c = \text{BayesGMM}(\dots, \theta_{1-k}^e)$,

$\mathcal{X}_k$ is calculated with $\mathcal{X}_k = \{\dots | \dots \in (X, Y', W_k^c)\}$,

θ_k^{e+1} is calculated with $\theta_k^{e+1} = \text{MixMatch}(\mathcal{X}_{1-k}, \dots, \theta_k^e)$

end for

// Stage III: Predictions using the pair DNNs

$$y = \arg\max \left(f_{\theta_1^E}(x) + f_{\theta_2^E}(x) \right)$$

$f_{\theta_1^E}(x)$ represents the prediction (output) of the first network

$f_{\theta_1^E}(x) + f_{\theta_2^E}(x)$ the idea is that you are combining the two networks' outputs to make a joint decision

Table 8.9 BadLabel

BadLabel
Input: A clean set $D = \{(x_i, y_i)\}_{i=1}^n$, flipping ratio ρ, iteration T, step size α

Given a C-class training set $D = \{(x_i, y_i) | x_i \in \mathbb{R}^d, y_i \in \{0, \dots, C-1\}\}_{i=1}^n$ where y_i is the clean label of x_i

Output: A label-noise set $D' = \{(x_i, y_i')\}_{i=1}^n$

Given a clean training set D, we flip $(100 \times \rho)\%$ of the clean labels that maximize the loss ℓ, which is formulated as follows

$$\mathop{\mathbb{E}}_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \left\{ \max_{y_i'} \ell\left(y_i', f(x_i)\right) \right\}$$

$$\text{s.t.} \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{y_1\}}(y_i') = 1 - \rho$$

where $y_i' \in \{0, \dots, C-1\}$, $\mathbb{1}_{\{\cdot\}}(\cdot)$ is the indicator function that ensures the designated label flipping ratio of clean labels

The algorithm goal is to find the modified label y_i' that maximize the loss $\ell\left(y_i', f(x_i)\right)$. This means you are exploring how poorly the model f predicts for each data point across all potential modified labels

Process

// Stage I: Optimize the data's affinity score $\boldsymbol{z(i,j)}$

Initialize flag array $z_1 \in \mathbb{R}^{n \times C}$ by Y (i.e., one-hot version of n clean label y_i).

for epoch $t = 1, \dots, T$ do

After each epoch, the model f_t gets better at predicting the clean labels y_i. However, as the model improves, it may start to memorize easier examples, while struggling with harder examples.

Iterate $\boldsymbol{D}$ to optimize $\boldsymbol{f_t}$

Updating f_t iteratively helps the algorithm progressively identify harder examples, which the model continues to struggle with

$$\arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i))$$

f denotes a classifier (i.e., a DNN), $\mathcal{F}$ is the hypothesis space, ℓ is the loss function for optimization

$$\ell\left(y_i, f(x_i)\right) = L = \{l_1, \dots, l_N\}^T$$

$$l_n = -\log \frac{\exp(x_n, y_n)}{\sum_{c=1}^{C} \exp\left(x_n, c\right)}$$

where x is the input, y is the target, C is the number of classes, N is the number of samples

Update $\boldsymbol{z_{t+1}}$

$$z_{t+1} = z_t - \alpha \nabla_Y \ell(Y, f_t(X))$$

At every epoch t, we update the flag array z,

where X is an $n \times d$ tensor (i.e., $x_1, \dots, x_n]^T$), Y is an $n \times C$ array (i.e., one-hot version of hard labels), $t \in \{1, 2, \dots, T\}$ is the iteration index, f_t is a DNN at epoch t, α is a small step size

Normalize $\boldsymbol{z_{t+1}}$ // e.g. use SoftMax function

$$\text{softmax}\big(z_{t+1}(i,j)\big) = \frac{e^{z_{t+1}(i,j)}}{\sum_{k=1}^{C} e^{z_{t+1}(i,k)}}$$

where $z_{t+1}(i,j)$ is the affinity score for the i-th data point and j-th class at epoch $t+1$, C is the total number of classes

The denominator sum the exponentiated affinity scores across all classes for the same data point i, ensuring that the resulting probabilities sum to 1

end for

// Stage II: Obtain $\boldsymbol{z_T}$ and flip $\boldsymbol{\rho}$ ratio of labels

Re-arrange $\boldsymbol{D}$ in ascending order by $\{\min \boldsymbol{z_T}(\boldsymbol{i},:)\}_{\boldsymbol{i=1}}^{\boldsymbol{n}}$

Select the first $\boldsymbol{\rho}$ percentage of data

for epoch $i = 1, \dots, [\rho \times n]$ do

Flip its label to the class with the lowest affinity score

$$y_i' = \arg\min z_T(i,:)$$

end for