applied sciences

MDPI

Article

Gradient Agreement Hinders the Memorization of Noisy Labels

Shaotian Yan [1], Xiang Tian [2], Rongxin Jiang [2] and Yaowu Chen [3,*]

[1] College of Biomedical Engineering and Instrument Science, Zhejiang University, Hangzhou 310007, China
[2] Zhejiang Provincial Key Laboratory for Network Multimedia Technologies, Hangzhou 310007, China
[3] Zhejiang University Embedded System Engineering Research Center, Institute of Advanced Digital Technologies and Instrumentation, Ministry of Education of China, Hangzhou 310007, China
* Correspondence: cyw@mail.bme.zju.edu.cn

Abstract: The performance of deep neural networks (DNNs) critically relies on high-quality annotations, while training DNNs with noisy labels remains challenging owing to their incredible capacity to memorize the entire training set. In this work, we use two synchronously trained networks to reveal that noisy labels may result in more divergent gradients when updating the parameters. To overcome this, we propose a novel co-training framework named gradient agreement learning (GAL). By dynamically evaluating the gradient agreement coefficient of every pair of parameters from two identical DNNs to determine whether to update them in the training process. GAL can effectively hinder the memorization of noisy labels. Furthermore, we utilize the pseudo labels produced by the two DNNs as the supervision for the training of another network, thereby gaining further improvement by correcting some noisy labels while overcoming the confirmation bias. Extensive experiments on various benchmark datasets demonstrate the superiority of the proposed GAL.

Keywords: noisy labeled data; robust learning; gradient methods; image classification

Citation: Yan, S.; Tian, X.; Jiang, R.; Chen, Y. Gradient Agreement Hinders the Memorization of Noisy Labels. *Appl. Sci.* **2023**, *13*, 1823. https://doi.org/10.3390/app13031823

Academic Editors: Andrea Prati, Luis Javier García Villalba and Vincent A. Cicirello

Received: 5 November 2022
Revised: 24 January 2023
Accepted: 27 January 2023
Published: 31 January 2023

1. Introduction

The performance of deep neural networks (DNNs) still largely depends on the quality of annotations despite the tremendous success they have achieved in a variety of visual tasks. There is a series of works on the learning mechanism of DNNs with noisy labels [1–3]. It is observed in [3] that DNNs are able to perfectly fit a randomly labeled training set owing to their strong memorization ability. When trained on a noisy dataset, the generalization performances of DNNs decrease sharply [2,4], due to overfitting of noisy labels. However, the accurate labeling of large scale datasets is almost impractical [5]. On one hand, it is extremely time consuming and costly to obtain high-quality labels for large scale datasets. Researchers tend to collect data and the labels automatically using social media or online search engines as a cheaper alternative, which inevitably introduces noisy labels. On the other hand, annotators need specific expertise to label some datasets (e.g., medical or agricultural datasets), which will easily produce incorrect labels caused by variability in the labeling by several annotators.

Therefore, noisy label learning has attracted increasing research interest in recent years [6–18]. Recent studies have demonstrated that co-training is conducive to noisy label learning. To hinder the memorization of noisy labels, MentorNet [6] trains a student network by feeding in small-loss samples selected by a teacher model because the prediction loss of true-labeled samples tends to be smaller than that of noisy samples [1]. Decoupling [7] trains two DNNs and only uses samples with divergent predictions from the two models to update them. Each network in Co-teaching [8] provides small-loss samples to the other. Co-teaching+ [9] gains further improvement by only selecting small-loss samples from those with different prediction labels from two models. By excluding samples with low certainty of being clean from the training, these methods can effectively avoid overfitting of mislabeled samples, yet they are criticized for leaving a large part of the training

dataset unused. JoCoR [10] jointly trains two networks using a joint loss consisting of cross entropy losses from each model and a KL-Divergence loss [19] measuring the distance of the two prediction logits. Co-learning [11] trains a shared feature encoder using a noisy label supervised classifier head along with a self-supervised projection head.

The two parallelly trained models in the above methods will gradually become in agreement with each other in the training process. With the constraining of KL-Divergence loss, JoCoR archives agreement on model predictions while Co-learning maximizes the agreement in latent space. In this study, we innovatively propose to alleviate the memorization of noisy labels by utilizing agreements on parameter gradients.

First, we conduct contrast experiments to explore how noisy labels affect the training process of DNNs. We train two identical networks with the same initialization. A gradient agreement coefficient is designed to evaluate whether the updating directions of two corresponding parameters from the two networks agree at every gradient descent step. Figure 1 presents the proportion of parameters that achieve agreements among all parameters under different levels of label noise on CIFAR-10. In every batch of the warm up period, the two networks are trained on exactly same input data. Therefore, as shown, the gradient direction of all parameter pairs are in total agreement. After that, we feed data with same class distributions but not consists of same samples to each network in every batch. As Figure 1 illustrates, the agreement ratio gradually decreases and finally reaches convergence. The higher the noisy rate is, the lower final agreement ratio becomes. While the decreases of agreement ratio occurring on original CIFAR-10 are mainly resulted by intra-class variance, the significant drops of final agreement ratios on other settings are obviously caused by noisy labels, indicating that noisy labels may result in divergent gradient directions in the late stage of training, which ultimately degrades the generation abilities of DNNs.

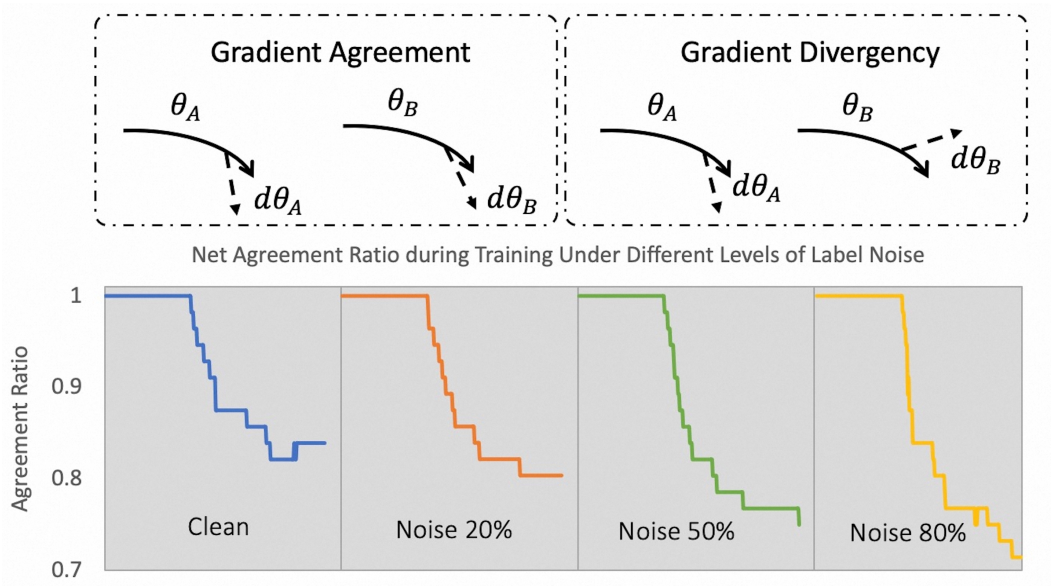

Figure 1. Experiments of gradient agreement using a pair of identical networks. We define parameters whose gradient directions achieve agreements with corresponding parameters from the other network as clean parameters. Agreement Ratio refers to the proportion of clean parameters among the whole network. The four curves depict, from left to right, the trend of agreement ratios in the whole training process under origin CIFAR-10 and CIFAR-10 with 20%, 50%, 80% noisy labels. X-axis is the number of iterations. In every batch of the early training period, the two nets are trained on exactly same input data. Therefore, the gradient direction of all parameter pairs are in total agreement. Later, batches with same class distributions but not consists of same samples are fed into each network respectively and the agreement ratio gradually decreases.

This phenomenon inspires us to hinder the memorization of noisy labels by identifying divergent gradient directions. However, besides noisy labels, there are several factors which will result in divergence of gradient directions, including different initialization and different class distribution of input samples. To distinguish the divergences caused by noisy labels, we need to eliminate the impact of other factors. Naturally, we propose a novel gradient agreement learning framework (GAL for short). GAL synchronously trains two identical networks with same initialization to avoid the gradient divergence caused by different initialization. Moreover, we propose class distribution agreement sampling strategy to ensure the samples input to the two networks in each iteration share same distribution of annotation classes while avoiding them being identical, thus eliminating the gradient divergence caused by different class distribution of input samples. Therefore, by excluding divergent gradient updates after the warm up steps, GAL can effectively prevent networks from memorizing noisy samples. Then, in every epoch, we use the prediction results of the two networks as pseudo-labels to supervise the training of a third net. In this way, we can rectify noisy labels while avoiding the accumulated confirmation biases.

Compared with previous co-training methods, GAL trains models on the entire training set and gains additional information from hard or noisy samples which have a great possibility not to be selected by small-loss methods such as Co-teaching+. Moreover, by seeking agreement on gradient directions, GAL still allows divergence on the final predictions from the two nets, thus obtaining more significant improvements from model ensembling.

In summary, the contributions of this paper is threefold:

- Contrast experiments show that noisy labels may cause more divergent gradients in the late stage of training.
- We propose a simple yet effective gradient agreement learning framework that effectively hinders the memorization of noisy labels.
- Extensive experiments show the effectiveness and robustness of GAL under different ratios and types of noise, outperforming previous methods.

2. Related Work

We briefly introduce existing literature on noisy label learning in this section.

Regularization. Regularization methods are widely used in the literature and are proved to be able to upgrade the generalization abilities of deep learning models. Recent studies [12–18] have proposed various regularization methods to prevent the memorization of noisy labels. Menon [15] design a new approach to clip gradients, which is robustness to noisy labels. Robust early-learning [16] dynamically divides parameters into critical and non-critical ones based on their importance for the learning of clean labels. Then different update strategies are applied to the two groups of parameters to avoid overfitting. However, noise rate is needed to divide the parameters, which is not available in real world cases. ELR [17] proposes an noise robust regularization term to steer models towards label probabilities produced based on model outputs. Label Smoothing [18] is a technique to train models with an estimated label distribution instead of the one-hot label, thereby hindering the memorization of noisy labels.

Co-training Methods. Recently, kinds of co-training methods for noisy label learning have been developed by researchers. Decoupling [7] proposes the "disagreement" strategy, only updating two simultaneously trained models based on samples with divergent predictions from the two networks. MentorNet [6] trains a student network using small-loss samples selected by a cooperating teacher model. Co-teaching [8] parallelly trains a pair of networks and updates them with small-loss samples selected by the peers. Co-teaching+ [9] then introduce the "disagreement" strategy into the training process of Co-teaching. In contrast, JoCoR [10] proposes to select confident samples with agreed predictions. Co-learning [11] tries to introduce more views of training data by training a shared feature encoder using a noisy label supervised classifier head along with a self-supervised projection head. By seeking agreements in latent space, models become tolerant to noisy

labels. Different from previous co-teaching kind methods which select samples with small loss yet leaving a large part of training set unused, the proposed GAL trains models on the whole training set by only excluding parameters with disagreed gradients from updating instead of excluding all the large loss samples, thereby benefiting from more supervisory signals. Moreover, while the previous co-training methods either achieve agreement on the output predictions (e.g., Co-teaching+ and JoCoR) or seek to maximize the agreement in latent space (e.g., Co-learning), GAL seeks agreement on gradient directions of each parameter pairs and still allows divergence on the final predictions from the two nets, thus obtaining more benefits in test accuracy from model ensembling.

Semi-supervised and self-supervised learning. With the rapid developments in the area of Semi-supervised [20] and Self-supervised learning [21], recent studies [22–26] have leveraged these techniques in noisy label learning. DivideMix [24] categorizes training samples into clean and noisy sets utilizing a two-component Gaussian Mixture Model [27]. Then semi-supervised learning is applied to train the networks by treating the clean set and noisy set as labeled and unlabeled set respectively. MOIT+ [25] employs supervised contrastive learning to pretrain the models. With the learned representations, samples are divided into clean or noisy sets, after which semi-supervised learning is applied to train a classifier. Sel-CL+ [26] utilizes the low-dimensional features pretrained with unsupervised contrastive learning to select confident pairs of samples for the supervised contrastive training of models, which enables the training of models to benefit from not only the pairs with correct annotations, but also the pairs which are mislabeled from the same class. Despite the promising classification accuracy achieved by these methods, their improvements mainly owe to the strong abilities of semi-supervised and self-supervised learning techniques which partly or completely ignore the annotation labels, thereby not providing new approaches in how to hinder the memorization of noisy labels in supervised learning. In contrast, the proposed method tries to alleviate the impact of noisy labels in the context of supervised learning using the annotations as input.

3. Proposed Method

In this section, we introduce GAL, our proposed framework for noisy label learning, in detail. We utilize a pair of identical network to distinguish noisy gradients and prevent the memorization of noisy labels, thus upgrading the performance of the trained models. The overview of GAL is illustrated in Figure 2. We propose a triplet network structure (i.e., θ_A, θ_B and θ_C in Figure 2). To avoid divergence caused by different initialization, θ_A and θ_B are two identical nets, while θ_C can be any classification network. At each mini-batch after warm-up period, training samples with same class distribution are fed into θ_A and θ_B. To hinder the memorization of noisy labels, for every pair of corresponding parameters, we evaluate whether their gradient directions achieve agreement and only update agreed parameters. To further improve the performance and avoid the accumulated confirmation bias, at every epoch, the predictions of θ_A and θ_B on the training set are used as pseudo labels to supervise the training of θ_C. In general, GAL consists of three parts: class distribution agreement sampling, gradient agreement updating and pseudo-labels supervising. The three components will be introduced in order in the following subsections.

3.1. Class Distribution Agreement Sampling

We present class distribution agreement sampling in this section. In order to avoid gradient disagreement caused by different input data, the samples input to θ_A and θ_B should share same class distribution in every iteration. However, if the samples are identical, the gradient directions for all the parameters of θ_A and θ_B will be exactly the same, resulting in failure to distinguish gradient disagreement caused by noisy labels. Thus, we propose a class distribution agreement sampling strategy to ensure roughly the same class distribution for the samples input to θ_A and θ_B in every batch while avoiding them being identical.

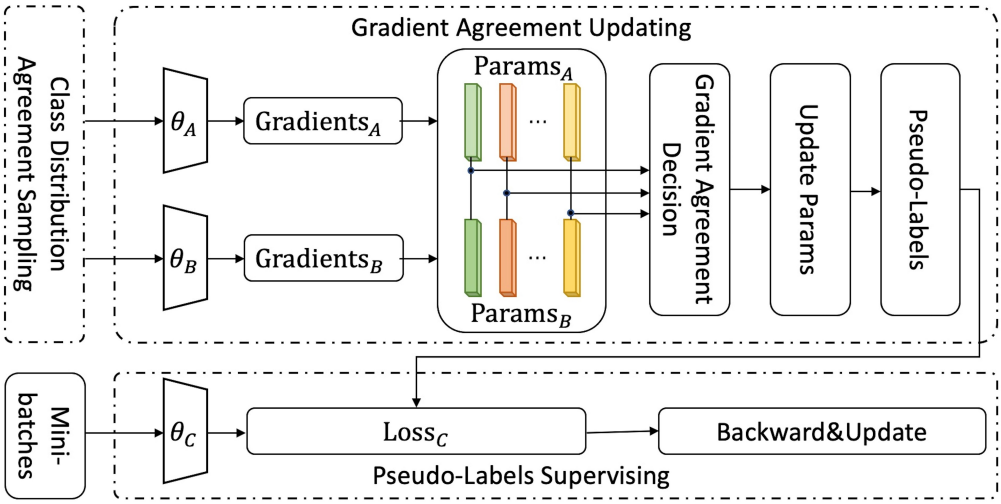

Figure 2. Overview of the proposed Gradient Agreement Learning (GAL) framework.

Let batches input to θ_A and θ_B in the nth iteration as $\mathcal{X}_n^a = \{(x_i^a, y_i^a)\}_{i=1}^{bs}$ and $\mathcal{X}_n^b = \{(x_i^b, y_i^b)\}_{i=1}^{bs}$ with bs being the batch size. As Figure 3 presents, the sampling process is repeated every four iterative steps. In the first iterative step, $\mathcal{X}_n^a$ is randomly sampled from the whole training set S with K annotation classes. Naturally, the annotated class distribution $\mathcal{D}_n^a$ of $\mathcal{X}_n^a$ can be obtained:

$$C_j = \{(y_i^a == j)\}_{i=1}^{bs}; \mathcal{D}_n^a = \{C_j\}_{j=1}^{K} \tag{1}$$

where C_j represents the number of samples belonging to class j in $\mathcal{X}_n^a$. Then $\mathcal{X}_n^b$ is formed by randomly sampling C_j instances for every class j in $\mathcal{D}_n^a$ among training samples except those in $\mathcal{X}_n^a$. In the second iteration, we feed $\mathcal{X}_n^b$ into θ_A and $\mathcal{X}_n^a$ into θ_B to make sure the training samples input to θ_A and θ_B are the same in one epoch. In the next two steps, to balance the training process of θ_A and θ_B, $\mathcal{X}_{n+2}^b$ is randomly sampled from S while $\mathcal{X}_{n+2}^a$ follows the class distribution of $\mathcal{X}_{n+2}^b$.

3.2. Gradient Agreement Updating

Algorithm 1 briefly describes Gradient Agreement Updating. The training procedure of θ_A and θ_B can be divided into two phases. Due to the large random volatility of the gradient directions at the early stage of training process, gradient agreement updating is not applied in the first phase. Meanwhile, previous works [2,3] have observed that DNNs tend to fit training samples with clean labels before memorizing noisy labels. Thus, in the first phase, we warm up the two models using the standard cross-entropy loss and gradient descent for a certain number of epochs. At this stage, parameters of the two identical nets θ_A and θ_B are initialized with the same values. The data fed into θ_A and θ_B is also identical in every batch to avoid unnecessary gradient divergence.

In the next phase, the two nets will gradually over-fit to noisy labels if following the training method in the first phase. Therefore, to hinder the memorization of noisy labels, gradient agreement updating is applied. θ_A and θ_B each consists of N parameters. We can group the corresponding parameters of the two networks to obtain N parameter pairs $\rho_i = \{\mathcal{P}_i^A, \mathcal{P}_i^B\}; i \in (1, \ldots, N)$. In every iteration, using cross entropy loss and the annotation labels, a pair of gradients $\nabla\mathcal{P}_i^A$ and $\nabla\mathcal{P}_i^B$ for ρ_i is obtained after each backward

propagation. In contrast to standard gradient descent methods which directly add gradients to ρ_i, an intermediate parameter pair $\hat{\rho}_i$ is first calculated following Equation (2):

$$\begin{aligned} \hat{\mathcal{P}}_i^A &= \mathcal{P}_i^A + \eta \nabla \mathcal{P}_i^A \\ \hat{\mathcal{P}}_i^B &= \mathcal{P}_i^B + \eta \nabla \mathcal{P}_i^B \end{aligned} \tag{2}$$

where η is the learning rate. We then reshape $\hat{\mathcal{P}}_i^A$ and $\hat{\mathcal{P}}_i^B$ to D-dimension vectors $\mathbf{v}_i^A$ and $\mathbf{v}_i^B$, respectively.

$$\begin{aligned} \mathbf{v}_i^A &= Reshape(\hat{\mathcal{P}}_i^A), \mathbf{v}_i^A \in (1, D) \\ \mathbf{v}_i^B &= Reshape(\hat{\mathcal{P}}_i^B), \mathbf{v}_i^B \in (1, D) \end{aligned} \tag{3}$$

To measure whether the gradients are clean or noisy, we propose a gradient agreement coefficient g_i which is defined as follows:

$$g_i = \Phi(\mathbf{v}_i^A, \mathbf{v}_i^B) = \frac{(\mathbf{v}_i^A)^{\mathbf{T}} \mathbf{v}_i^B}{\sqrt{\Sigma_{j=1}^{\mathbf{D}} (\mathbf{v}_{ij}^A)^2} \times \sqrt{\Sigma_{j=1}^{\mathbf{D}} (\mathbf{v}_{ij}^B)^2}} \tag{4}$$

After acquiring g_i, we then apply parameter updating as follows:

$$\begin{aligned} \widetilde{\mathcal{P}}_i^A &= \mathcal{P}_i^A + \mathbb{1}_{g_i \geq \tau} * \eta \nabla \mathcal{P}_i^A \\ \widetilde{\mathcal{P}}_i^B &= \mathcal{P}_i^B + \mathbb{1}_{g_i \geq \tau} * \eta \nabla \mathcal{P}_i^B \end{aligned} \tag{5}$$

where τ is the threshold for determining whether the gradients of a pair of parameters reach an agreement.

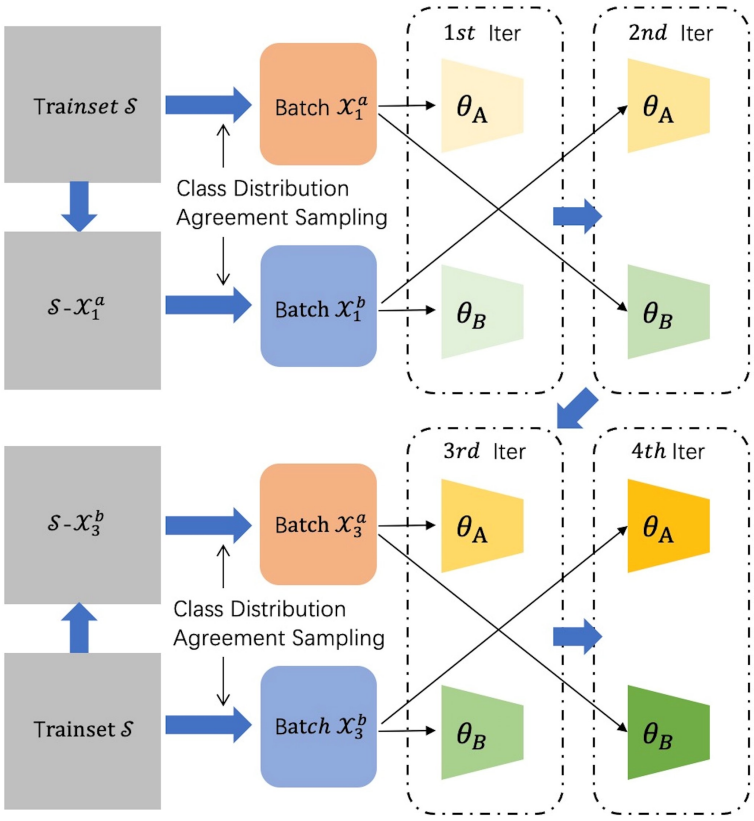

Figure 3. Class Distribution Agreement Sampling. $\mathcal{X}_1^a$ and $\mathcal{X}_3^b$ are randomly sampled from S while $\mathcal{X}_1^b$ and $\mathcal{X}_3^a$ from $S - \mathcal{X}_1^a$ and $S - \mathcal{X}_3^b$ respectively. $\mathcal{X}_1^a$ and $\mathcal{X}_1^b$ share same annotated class distribution, so are $\mathcal{X}_3^a$ and $\mathcal{X}_3^b$.

Algorithm 1: Gradient Agreement Updating.

Input: θ_A, θ_B, θ_C, *param update threshold* τ, *learning rate* η, *batch size bs*, *training dataset* $\mathcal{S} = (\mathcal{X}, \mathcal{Y}) = \{(x_i, y_i)\}_{i=1}^{N}$

for *e = 0 to MaxEpoch* **do**
  /*warm up θ_A and θ_B for certain epochs*/
  if $e <$ *WarmUpEpoch* **then**
    θ_A = WarmUp($\mathcal{S}$,θ_A)
    θ_B = WarmUp($\mathcal{S}$,θ_B)
  else
    for *iter = 1 to IterNum* **do**
      Draw two mini-batches $\{(x_n^a, y_n^a)\}_{n=1}^{bs}$ and $\{(x_n^b, y_n^b)\}_{n=1}^{bs}$ from $\mathcal{S}$ using Class Distribution Agreement Sampling strategy
      /*performing a forward propagation step*/
      $\mathcal{L}_a = \frac{1}{bs} CE(\mathcal{P}_{model}(x_n^a; \theta_A), y_n^a)_{n=1}^{bs}$
      $\mathcal{L}_b = \frac{1}{bs} CE(\mathcal{P}_{model}(x_n^b; \theta_B), y_n^b)_{n=1}^{bs}$
      /*propagating the loss back to obtain gradients of parameters*/
      Backward($\mathcal{L}_a$)
      Backward($\mathcal{L}_b$)
      for $\mathcal{P}_i^A, \mathcal{P}_i^B$ *in* (θ_A, θ_B) **do**
        /*add gradients to parameters*/
        $\hat{\mathcal{P}}_i^A = \mathcal{P}_i^A + \eta \nabla \mathcal{P}_i^A$
        $\hat{\mathcal{P}}_i^B = \mathcal{P}_i^B + \eta \nabla \mathcal{P}_i^B$
        /* compute gradient agreement coefficient g_i */
        $g_i = \Phi(Reshape(\hat{\mathcal{P}}_i^A), Reshape(\hat{\mathcal{P}}_i^B))$
        /* compare g_i to threshold to determine whether to update the parameters */
        $\widetilde{\mathcal{P}}^{A_i} = \mathcal{P}^{A_i} + \mathbb{1}_{g_i \geq \tau} * \eta \nabla \mathcal{P}^{A_i}$
        $\widetilde{\mathcal{P}}^{B_i} = \mathcal{P}^{B_i} + \mathbb{1}_{g_i \geq \tau} * \eta \nabla \mathcal{P}^{B_i}$
      end
    end
  end
end

3.3. Pseudo-Labels Supervising

The training set $\mathcal{S} = (\mathcal{X}, \mathcal{Y})$ consists of input data $\mathcal{X}$ and corresponding labels $\mathcal{Y}$. Part of $\mathcal{Y}$ is made up of noisy labels. Models will over-fit to noisy labels if directly using $\mathcal{Y}$ as the supervision for training, downgrading the generalization abilities. The gradient agreement updating method proposed above can effectively prevent θ_A and θ_B from memorizing noisy labels, yet the performance of trained models can be further strengthened leading by more correct supervision. However, if we directly correct the labels used to supervising θ_A and θ_B, the training of models will suffer from accumulated confirmation biases. Therefore, we propose to train a third net θ_C as shown in Algorithm 2.

First, we use the average of the prediction logits outputted by θ_A and θ_B for sample x_i to represent the joint prediction:

$$p_i^{joint} = \frac{1}{2}(\mathcal{P}_{model}(x_i; \theta_A) + \mathcal{P}_{model}(x_i; \theta_B)) \tag{6}$$

Then the label with maximum score in p_i^{joint} is used as pseudo-label $\hat{y}_i$:

$$\hat{y}_i = \arg\max(p_i^{joint}) \tag{7}$$

With the logit p_i^c predicted by θ_C for sample x_i according to Equation (8), the training of θ_C is supervised by two losses, named classification loss and prediction logit loss respectively.

$$p_i^c = \mathcal{P}_{model}(x_i; \theta_C) \tag{8}$$

Because θ_A and θ_B are hindered from memorizing noisy labels, the precision of pseudo labels generated is apparently much higher than the original annotation labels. Therefore, the classification loss $\mathcal{L}_c$ is defined as:

$$\mathcal{L}_c = \frac{1}{bs} \sum_{i=1}^{bs} \mathbb{1}_{max(p_i^{joint}) > \epsilon} CE(p_i^c, \hat{y}_i) \tag{9}$$

where only pseudo labels with confidence scores higher than ϵ will be counted in the loss.

Algorithm 2: Pseudo-Labels Supervising.

Input: θ_A, θ_B, θ_C, *confidence threshold* ϵ, *batch size* bs, *class number* M, *training dataset* $\mathcal{S} = (\mathcal{X}, \mathcal{Y})$

for e = *WarmUpEpoch to MaxEpoch* **do**
- Draw a mini-batch $(x_i, y_i)_{i=1}^{bs}$ from $\mathcal{S}$
- /* obtain the joint prediction logit p^{joint_i} for sample x_i */
- $p_i^{joint} = \frac{1}{2}(\mathcal{P}_{model}(x_i; \theta_A) + \mathcal{P}_{model}(x_i; \theta_B))$
- /* generate pseudo label $\hat{y}_i$ for sample x_i */
- $\hat{y}_i = \arg\max(p_i^{joint})$
- /* obtain the prediction logit p_i^c for sample x_i using θ_C */
- $p_i^c = \mathcal{P}_{model}(x_i; \theta_C)$
- /* classification loss for θ_C */
- $\mathcal{L}_c = \frac{1}{bs} \sum_{i=1}^{bs} \mathbb{1}_{max(p_i^{joint}) > \epsilon} CE(p_i^c, \hat{y}_i)$
- /* prediction logit loss for θ_C */
- $\mathcal{L}_d = \frac{1}{bs} \sum_{i=1}^{bs} \sum_{j=1}^{M} p_{ij}^{joint} \log \frac{p_{ij}^{joint}}{p_{ij}^c}$
- /* update θ_C with the joint loss */
- $\mathcal{L} = \mathcal{L}_d + \mathcal{L}_c$

end

The prediction logit loss $\mathcal{L}_d$ is defined as the Kullback–Leibler Divergence between the joint logits and the output prediction logits of θ_C:

$$\mathcal{L}_d = \frac{1}{bs} \sum_{i=1}^{bs} \sum_{j=1}^{M} p_{ij}^{joint} \log \frac{p_{ij}^{joint}}{p_{ij}^c} \tag{10}$$

Then, the overall loss $\mathcal{L}$ of θ_C is the sum of $\mathcal{L}_d$ and $\mathcal{L}_c$:

$$\mathcal{L} = \mathcal{L}_d + \mathcal{L}_c \tag{11}$$

Leading by $\mathcal{L}_d$, θ_C is effectively prevented from over-fitting to noisy labels. Meanwhile the performance of θ_C is further enhanced with $\mathcal{L}_c$ which brings in more correct supervision.

4. Experiments

4.1. Experiment Settings

4.1.1. Datasets

GAL is validated on various popular benchmark datasets for learning from noisy labels. CIFAR-10 [28] and CIFAR-100 [29], both of which contain 50 K training images and 10 K

test images, are accurately labeled. To maintain consistency with previous works [10,11], we evaluate two types of synthetic noisy labels on the two datasets. In symmetric noise (Sym for short) settings, every category has the same probability to be labeled as a random class. As for asymmetric noise (Asym), a certain percentage of labels of each class are relabeled into a visually similar category (e.g., cat $\rightarrow$ dog).

Animal-10N [30] consists of 50 K human-labeled training images with an estimated noise level of 8% and 5 K clean testing images. The images are collected online, belonging to 10 classes of animals.

Clothing1M [31] is made up of 1 million training images with labels generated from website descriptions and an accurately labeled validation with about 10 K images. Due to the collecting methods of its training samples, the noise rate of Clothing1M is approximately 38.5% [32].

4.1.2. Implementation Details

CIFAR-10 and CIFAR-100. For experiments on CIFAR-10 and CIFAR-100, an 18-layer PreAct Resnet [33] is adopted for θ_A, θ_B and θ_C. The training process takes 200 epochs with a batch size of 128. The initial learning rate is 0.06 and is reduced by a factor of 10 after 80,150 epochs. The warm-up period is set to 80 epochs all the experiments on CIFAR-10 and CIFAR-100. The experiments are conducted on a single NVIDIA Tesla V100 GPU.

Animal-10N. We employ ResNet-34 [34] as the backbones of the three networks to evaluate GAL on Animal-10N. The training process lasts 150 epochs using a single NVIDIA Tesla V100 GPU with a mini-batch size of 128. We set the initial learning rate as 0.01, and reduce it by a factor of 10 after 60, 90, and 120 epochs. The warm-up period consists of 60 epochs.

Clothing1M. For experiments on Clothing1M, an imagenet pretrained resnet-50 [34] is adopted as the backbone of the three networks. The training procedure lasts 40 epochs with a batch size of 32 using four NVIDIA Tesla V100 GPUs. The initial learning rate is 0.01 and is reduced by a factor of 10 after 10, 20 and 30 epochs. The warm-up period is set to 10 epochs.

In all the experiments, we train all three nets using SGD with a momentum of 0.9 and a weight decay of 0.0005. The gradient agreement threshold τ is set to 0.4 and the pseudo-label threshold ϵ is set to 0.8. The reported results of GAL in Tables 1–3 are obtained only using the prediction of θ_C.

4.2. Compared Methods

In this section, we compare the performance of the proposed GAL with previous methods on the four popular benchmark datasets. Table 1 compares the performance of the proposed GAL and previous co-training methods on CIFAR-10 and CIFAR-100 with synthetic noisy labels. Table 2 presents the evaluation results on Animal-10N with real-world noisy labels. It can be seen that GAL significantly outperforms the other methods in most experimental settings. In particular, there exists an obvious performance drop between the best and the last test accuracy for most methods, especially under high noise levels (e.g., 24.1 for JoCoR and 13.8 for co-learning under 80% symmetric noise rate on CIFAR-10). This implies that the models are over-fitted to noisy labels during the training.

In sharp contrast, while achieving better accuracy than other methods, the performances of GAL are quite stable under all types and ratios of noise on CIFAR-10 and CIFAR-100 with the maximum drop between the best and the last test accuracy being merely 0.6. By seeking agreement on gradient directions, GAL can effectively hinder the memorization of noisy labels. While the previous co-training methods either achieve agreement on the output predictions (e.g., Co-teaching+ and JoCoR) or seek to maximize the agreement in latent space (e.g., Co-learning), GAL still allows divergence on the final predictions from the two nets, thus obtaining more benefits in test accuracy from model ensembling. Moreover, the performance is further enhanced with the supervision of pseudo labels.

Table 1. Performance comparison with previous methods. Models are trained on CIFAR-10 and CIFAR-100 with different ratios and types of label noise as PreAct-18 resnet being the backbone and tested on a clean testing set. The best test accuracy in the whole training process and the test accuracy of the last epoch are reported.

Dataset		**CIFAR-10**				**CIFAR-100**		
Method/Noise Ratio		**Sym 20%**	**Sym 50%**	**Sym 80%**	**Asym 40%**	**Sym 20%**	**Sym 50%**	**Sym 80%**
Cross-Entropy	Best	86.8	79.4	62.9	85.0	62.0	46.7	19.9
	Last	82.7	57.9	26.1	72.3	61.8	37.3	8.8
	Best-Last	4.1	21.5	36.8	12.7	0.2	9.4	11.1
MentorNet [6]	Best	86.6	82.4	63.1	-	61.8	47.3	22.8
	Last	85.2	81.3	41.9	-	61.1	38.5	11.6
	Best-Last	1.4	1.1	21.2	-	0.7	8.8	11.2
Co-teaching [8]	Best	87.9	83.5	64.4	87.1	63.4	50.3	25.6
	Last	86.5	81.9	44.2	85.2	62.8	48.5	13.2
	Best-Last	1.4	1.6	20.2	1.9	0.6	1.8	12.4
Co-teaching+ [9]	Best	89.5	85.7	67.4	87.8	65.6	51.8	27.9
	Last	88.2	84.1	45.5	86.4	64.1	45.3	15.5
	Best-Last	1.3	1.6	21.9	1.4	1.5	6.5	12.4
JoCoR [10]	Best	90.2	85.3	69.2	88.3	65.4	52.5	27.1
	Last	88.9	83.8	45.1	86.6	64.3	50.4	15.7
	Best-Last	1.1	1.5	24.1	1.7	1.1	2.1	11.4
Co-learning [11]	Best	93.1	88.6	76.2	88.0	69.5	59.8	38.6
	Last	92.5	87.6	62.4	86.5	68.9	59.1	35.2
	Best-Last	0.6	1.0	13.8	1.5	0.6	0.7	3.4
GAL	Best	93.7	90.5	78.9	91.6	70.5	61.3	40.8
	Last	93.5	90.1	78.3	91.2	70.1	60.8	40.2
	Best-Last	0.2	0.4	0.6	0.4	0.4	0.5	0.6

Table 2. Performance comparison on Animal-10N.

Method	**CE**	**Decoupling**	**Co-Teaching**	**Co-Teaching+**	**JoCoR**	**Co-Learning**	**GAL**
Best	82.68	79.22	82.43	50.66	82.88	82.95	83.33
Last	81.1	78.24	81.52	48.52	81.06	81.18	82.91

Table 3. Performance comparison on clothing1M.

Method	**JoCoR**	**TCNet [35]**	**ELR [16]**	**FINE [36]**	**F-div [37]**	**Label Smooth [18]**	**GAL**
Accuracy	70.30	71.15	72.87	72.91	73.09	73.44	73.62

Table 3 presents the evaluation results on the real-world benchmark Clothing1M of GAL and several baseline methods, i.e., co-training method JoCoR, sample selection methods (TCNet, Fine) and regularization methods (ELR, F-div, Label Smoothing). GAL works well on Clothing1M, demonstrating its effectiveness in learning with noisy labels.

4.3. Ablation Study

4.3.1. Ablation Study on Components

GAL synchronously trains two networks with gradient agreement updating and uses pseudo-labels to supervise the learning of a third network. To evaluate the contribution of each part, we perform ablation studies in this section. The evaluation results are presented in Table 4. The experiment named "θ_A + θ_B + Cross Entropy" refers to training two nets using standard cross-entropy loss and using the average prediction results of the two models to calculate the accuracy. In the experiments named "Random Updating",

the network is warmed up with 80 epochs and then we randomly update a certain proportion of parameters in every iterations. In the experiments named "θ_A + θ_B + Random Updating", two nets are trained synchronously. After warming up, a certain proportion of parameter pairs in the two nets are randomly selected for updating. For the above two settings, 80%, 50%, 20% and 60% of parameters are randomly selected under 20%, 50%, 80% symmetric noise and 40% asymmetric noise respectively. As shown in Table 4, when applying gradient agreement updating to the two nets, the performances in different experimental settings sharply increase, manifesting its effectiveness. The performances of "θ_A + θ_B + Gradient Agreement Updating" are significantly higher than those of "θ_A + θ_B + Random Updating", further demonstrating the effectiveness of selecting parameter pairs based on gradient agreement. The full GAL method gains further improvements by training a third network supervised by pseudo-labels produced by the two jointly trained models. Notably, only the third net is used for inference when reporting the performance of the full GAL.

Table 4. Results of ablation study on CIFAR-10 with different types and levels of label noise

Dataset	**CIFAR-10**			
Method	**Sym 20%**	**Sym 50%**	**Sym 80%**	**Asym 40%**
Cross Entropy	86.8	79.4	62.9	85.0
Random Updating	83.1	74.2	56.5	81.8
θ_A + θ_B + Cross Entropy	88.9	83.6	63.9	86.5
θ_A + θ_B + Random Updating	83.4	74.3	57.1	82.2
θ_A + θ_B + Gradient Agreement Updating	93.5	89.8	77.7	91.1
GAL	93.7	90.5	78.9	91.6

4.3.2. Ablation Study on Agreement Sampling Methods

In this section, we perform ablation studies on different data sampling methods to verify the effectiveness of the proposed class distribution agreement sampling. With Random Sampling, instances fed into θ_A and θ_B in every batches are different and both randomly sampled from the whole training set. Models fail to learn from clean labels because the updates of parameters are hindered by gradient divergences caused by data difference. With Identical Sampling, exact same samples are fed into θ_A and θ_B in every iteration. Naturally, the gradients of the two models are always in agreement. Therefore, GAL fails to prevent models from memorizing noisy labels. Instead, the proposed class distribution agreement sampling strategy avoids gradient disagreement caused by different input data, meanwhile enables models to distinguish noisy gradients. As is shown in Table 5, the test accuracy is significantly higher utilizing class distribution agreement sampling. "CDAS w/o Reversed Sampling" refers to class distribution agreement sampling without reversed sampling from θ_B to θ_A. As Table 5 presents, the contribution of reversed sampling strategy to the performance of trained models is not very significant. However, by reducing the deviation of accuracy, it is conducive to stabilize the performance of GAL.

Table 5. Ablation Study on Agreement Sampling Methods. The mean accuracy and its standard deviation are computed over five tries.

Dataset	**CIFAR-10**			
Method	**Sym 20%**	**Sym 50%**	**Sym 80%**	**Asym 40%**
Random Sampling	84.9 $\pm$ 0.4	81.9 $\pm$ 0.6	65.2 $\pm$ 0.9	85.2 $\pm$ 0.2
Identical Sampling	86.8 $\pm$ 0.1	81.4 $\pm$ 0.1	65.1 $\pm$ 0.2	86.5 $\pm$ 0.1
CDAS w/o Reversed Sampling	93.2 $\pm$ 0.5	89.8 $\pm$ 0.8	78.1 $\pm$ 0.8	91.2 $\pm$ 0.4
Class Distribution Agreement Sampling	93.6 $\pm$ 0.2	90.3 $\pm$ 0.3	78.6 $\pm$ 0.4	91.5 $\pm$ 0.2

4.3.3. Ablation Study on Agreement Threshold

In this section, we analyse the impact of different values of gradient agreement threshold τ on the training process of GAL. Figure 4 presents the test accuracy acquired by the joint prediction of θ_A and θ_B on CIFAR-10 with different values of τ. When τ is set to zero, all the parameters of θ_A and θ_B will be updated without any constraint in every iterations. Therefore, the test accuracy occurs severe drop owing to memorize noisy labels. In the gradient agreement updating phase, data fed into θ_A and θ_B is not identical. Therefore, θ_A and θ_B can hardly be updated when τ is set to 1.0. As is shown in Figure 4, with other values of τ the models can effectively hinder the memorization of noisy labels and achieve optimal performance when τ is set to 0.4.

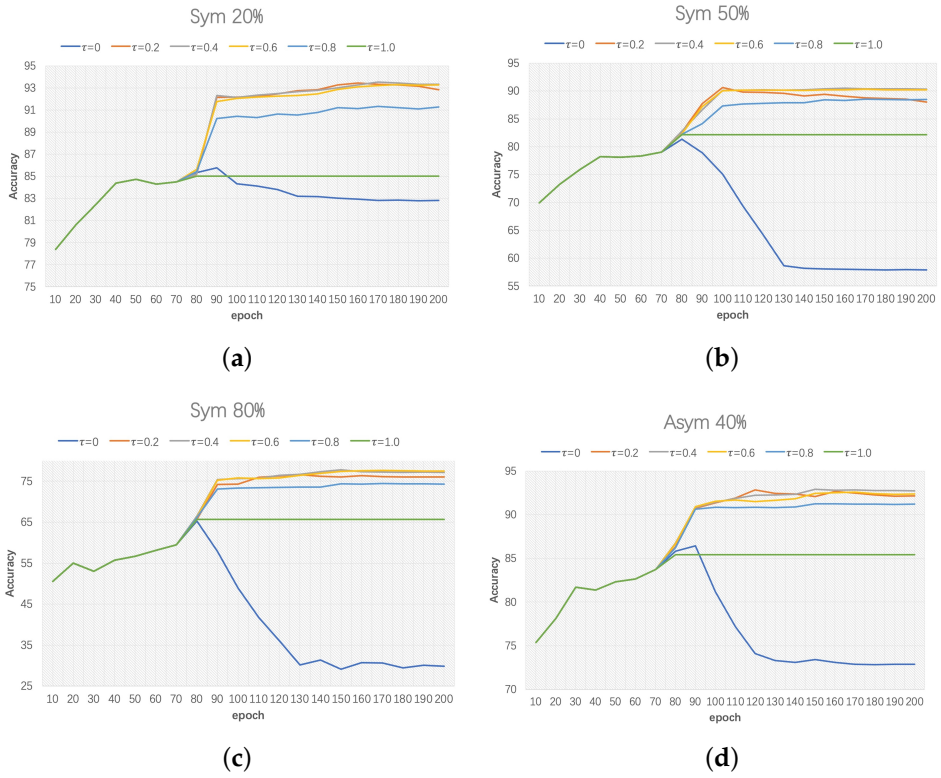

Figure 4. Analysis of different values of gradient agreement threshold τ on CIFAR-10 with different types and levels of label noise. (**a**) Test accuracy on 20% symmetric noise. (**b**) Test accuracy on 50% symmetric noise. (**c**) Test accuracy on 80% symmetric noise. (**d**) Test accuracy on 40% asymmetric noise.

4.4. Training Time Analysis

In this section, we compare the training time of GAL on CIFAR-10 with 50% symmetrical noise with previous methods. The results are listed in Table 6. We use a single Nvidia Tesla V100 GPU to train all the models. Cross Entropy is the fastest but its performance is not satisfying. The training time of Co-teaching and Co-teaching+ are almost the same. JoCoR is sightly slower than Co-teaching+. Although the three methods are both faster than Co-learning and GAL, the performances of Co-learning and GAL are much stronger. The training time of GAL is slightly shorter than Co-learning. In the warm-up period of GAL, the two networks with same initialization are trained with the same data in every iterations. Therefore, in practice, we just need to train one network in the warm-up period and clone it to form a pair of identical networks in the beginning of the second stage. At the second stage, despite the fact that GAL needs to calculate many parameters, all the calculations are undertaken on GPU. Consequently, the training time of GAL is still acceptable.Further more, the training process of GAL can be further shortened by speeding up the backward updating process with parallel computing.

Table 6. Comparision of total Training Time on CIFAR-10 with 50% symmetrical label noise, using a single Nvidia Telsa V100 GPU.

Cross Entropy	Co-Teaching	Co-Teaching+	JoCoR	Co-Learning	GAL
2.1 h	4.3 h	4.3 h	4.4 h	5.2 h	5.1 h

5. Conclusions and Future Work

In this study, using contrast experiments, we observe that noisy labels may cause more divergent gradients in the late stage of training. Thus, we propose a novel gradient agreement learning framework (GAL) to tackle the problem of learning with noisy labels. By synchronously training two nets and dynamically excluding divergent gradients, detected using a gradient agreement coefficient, from parameter updating, GAL is highly effective in hindering the memorization of noisy labels. Training a third network with the pseudo labels produced by the two nets further enhances the performance. Extensive experiments on CIFAR-10, CIFAR-100, Animal-10N and Clothing1M datasets demonstrate the effectiveness of GAL.

Limitations. Nowadays, the parameter size of deep neural networks is becoming more and more huge with the introducing of big models. Therefore, the training efficiency of GAL might become a drawback in training big models, with the gradient agreement coefficients for billions of parameter pairs needed to be calculated. Further studies should be conducted to reduce the computational complexity on big models.

Future Work. Our further work will focus on two aspects. (a) Reduce the training time of GAL on big models with deeper studying of the impact of noisy labels on different parameters. (b) Current noisy label learning techniques are mostly studied and applied in the task of image classification. Extending these works to other areas will be interesting.

Author Contributions: Conceptualization, S.Y. and X.T.; methodology, S.Y.; validation, S.Y., X.T. and R.J.; formal analysis, S.Y. and R.J.; writing—original draft preparation, S.Y.; writing—review and editing, X.T., R.J. and Y.C.; supervision, Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Fundamental Research Funds for the Central Universities.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this work included, i.e., CIFAR-10, CIFAR-100, Animal-10N and Clothing1M image sets, are openly available.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Arpit, D.; Jastrzebski, S.; Ballas, N.; Krueger, D.; Bengio, E.; Kanwal. M.S.; Maharaj, T.; Fischer, A.; Courville, A.; Bengio, Y.; et al. A closer look at memorization in deep networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 7–9 August 2017.
2. Toneva, M.; Sordoni, A.; des Combes, R. T.; Trischler, A.; Bengio, Y.; Gordon, G. An empirical study of example forgetting during deep neural network learning. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
3. Zhang, C.; Recht B.; Bengio, S.; Hardt, M.; Vinyals, O. Understanding deep learning requires rethinking generalization. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
4. Song, H.; Kim, M.; Park, D.; Shin, Y.; Lee, J.G. Learning from noisy labels with deep neural networks: A survey. *IEEE Trans. NNLS* **2022**. [CrossRef] [PubMed]
5. Bernhardt, M.; Castro, D.C.; Tanno, R.; Schwaighofer, A.; Tezcan, K.C.; Monteiro, M.; Bannur, S.; Lungren, M.P.; Nori, A.; Glocker, B.; et al. Active label cleaning for improved dataset quality under resource constraints. *Nat. Commun.* **2022**, *13*, 1161. [CrossRef] [PubMed]
6. Jiang, L.; Zhou, Z.; Leung, T.; Li, L.; Fei-Fei, L. MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018.

7. Malach, E.; Shalev-Shwartz, S. Decoupling "when to update" from "how to update". In Proceedings of the Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–7 December 2017.
8. Han, B.; Yao, Q.; Yu, X.; Niu, G.; Xu, M.; Hu, W.; Tsang, I.; Sugiyama, M. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In Proceedings of the Conference on Neural Information Processing Systems, Montreal, Canada, 2–8 December 2018.
9. Yu, X.; Han, B.; Yao, J.; Niu, G.; Tsang I.W.; Sugiyama, M. How does disagreement help generalization against label corruption? In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019.
10. Wei, H.; Feng, L.; Chen, X.; An, B. Combating noisy labels by agreement: A joint training method with co-regularization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Virtual, 14–19 June 2020.
11. Tan, C.; Xia, J.; Wu, L.; Li, S.Z. Co-learning: Learning from noisy labels with self-supervision. In Proceedings of the ACM International Conference on Multimedia, Chengdu, China, 20–24 October 2021.
12. Jenni S.; Favaro, P. Deep bilevel learning. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018.
13. Tanno, R.; Saeedi, A.; Sankaranarayanan, S.; Alexander, D. C.; Silberman, N. Learning from noisy labels by regularized estimation of annotator confusion. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019.
14. Hendrycks, D.; Lee, K.; Mazeika, M. Using pre-training can improve model robustness and uncertainty. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019.
15. Menon, A.K.; Rawat, A.S.; Reddi, S.J.; Kumar, S. Can gradient clipping mitigate label noise? In Proceedings of the International Conference on Learning Representations, Virtual, 26–30 April 2020.
16. Xia, X.; Liu, T.; Han, B.; Gong, C.; Wang, N.; Ge, Z.; Chang, Y. Robust early-learning: Hindering the memorization of noisy labels. In Proceedings of the International Conference on Learning Representations, Virtual, 3–5 May 2021.
17. Liu, S.; Niles-Weed, J.; Razavian, N.; Fernandez-Granda, C. Early-learning regularization prevents memorization of noisy labels. In Proceedings of the Conference on Neural Information Processing Systems, Virtual, 6–12 December 2020.
18. Lukasik, M.; Bhojanapalli, S.; Menon, A.; Kumar, S. Does label smoothing mitigate label noise? In Proceedings of the International Conference on Learning Representations, Virtual, 26–30 April 2020.
19. Kullback, S.; Leibler, R. On Information and Sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86. [CrossRef]
20. Yang, X.; Song, Z.; King, I.; Xu, Z. A Survey on Deep Semi-supervised Learning. *arXiv* **2021**, arXiv:2103.00550.
21. Liu, X.; Zhang, F.; Hou, Z.; Mian, L.; Wang, Z.; Zhang, J.; Tang, J. Self-supervised Learning: Generative or Contrastive. *IEEE Trans. KDE* **2021**, *35*, 857–876. [CrossRef]
22. Nguyen, D.T.; Mummadi, C.K.; Ngo, T.P.N.; Nguyen, T.H.P.; Beggel, L.; Brox, T. SELF: Learning to filter noisy labels with self-ensembling. In Proceedings of the International Conference on Learning Representations, Virtual, 26–30 April 2020.
23. Zhou, T.; Wang, S.; Bilmes, J. Robust curriculum learning: From clean label detection to noisy label self-correction. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 4 May 2021.
24. Li, J.; Socher, R.; Hoi, S. Dividemix: Learning with noisy labels as semi-supervised learning. In Proceedings of the International Conference on Learning Representations, Virtual, 26–30 April 2020.
25. Ortego, D.; Arazo, E.; Albert, P.; O'Connor, N.E. Multi-objective interpolation training for robustness to label noise. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Virtual, 20–25 June 2021.
26. Li, S.; Xia, X.; Ge, S.; Liu, T. Selective-Supervised Contrastive Learning with Noisy Labels. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 21–24 June 2022.
27. Permuter, H.; Francos, J.; Jermyn, I. A study of gaussian mixture models of color and texture features for image classification and segmentation. *Pattern Recognit* **2006**, *39*, 695–706. [CrossRef]
28. Krizhevsky, A.; Nair, V.; Hinton, G. *CIFAR-10*; Canadian Institute for Advanced Research: Toronto, ON, Canada, 2021.
29. Krizhevsky, A.; Nair, V.; Hinton, G. *CIFAR-100*; Canadian Institute for Advanced Research: Toronto, ON, Canada, 2021.
30. Song, H.; Kim, M.; Lee, J. SELFIE: Refurbishing Unclean Samples for Robust Deep Learning. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019.
31. Xiao, T.; Xia, T.; Yang, Y.; Huang, C.; Wang, X. Learning from massive noisy labeled data for image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 8–10 June 2015.
32. Song, H.; Kim, M.; Park D.; Lee, J. Prestopping: How does early stopping help generalization against label noise? In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019.
33. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016.
34. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26–30 June 2016.

35. Yi, R.; Huang, Y. TC-Net: Detecting Noisy Labels via Transform Consistency. *IEEE Trans. Multimed.* **2021**, *24*, 4328–4341. [CrossRef]
36. Kim, T.; Ko, J.; Choi, J.; Yun, S.Y. Fine samples for learning with noisy labels. In Proceedings of the Conference on Neural Information Processing Systems, Virtual, 7–10 December 2021.
37. Wei, J.; Liu Y. When Optimizing f-divergence is Robust with Label Noise. In Proceedings of the International Conference on Learning Representations, Virtual, 3–5 May 2021.