SYNERGISTIC NETWORK LEARNING AND LABEL CORRECTION FOR NOISE-ROBUST IMAGE CLASSIFICATION

Chen Gong[†1], *Kong Bin*[†2], *Eric J. Seibel*[1], *Xin Wang*[*2], *Youbing Yin*[*2], *Qi Song*[2]

[1]Mechanical Engineering, University of Washington, Seattle, WA 98195, USA
[2]Keya Medical, Seattle, WA 98104, USA

ABSTRACT

Large training datasets almost always contain examples with inaccurate or incorrect labels. Deep Neural Networks (DNNs) tend to overfit training label noise, resulting in poorer model performance in practice. To address this problem, we propose a robust label correction framework combining the ideas of small loss selection and noise correction, which learns network parameters and reassigns ground truth labels iteratively. Taking the expertise of DNNs to learn meaningful patterns before fitting noise, our framework first trains two networks over the current dataset with small loss selection. Based on the classification loss and agreement loss of two networks, we can measure the confidence of training data. More and more confident samples are selected for label correction during the learning process. We demonstrate our method on both synthetic and real-world datasets with different noise types and rates, including CIFAR-10, CIFAR-100 and Clothing1M, where our method outperforms the baseline approaches.

***Index Terms*—** Noise label, Image classification, Small loss selection, Label correction, Iterative learning

1. INTRODUCTION

Deep learning has shown very impressive performance on various vision problems. However, a practical challenge for deep learning state-of-the-art models is that they rely on large amounts of clean, annotated data [1, 2, 3, 4]. Collecting such data sets is expensive or time-consuming. Large training datasets almost always contain examples with inaccurate or incorrect labels, resulting in overfitting to noisy samples and poorer model performance [5, 6, 7, 8]. There are many classic methods to prevent noise overfitting such as dropout [9] and early stop [10], which are heuristic for noisy label learning.

A large number of algorithms have been developed for learning with noisy labels. Small loss selection recently achieved great success on noise-robust deep learning following the widely used criterion: DNNs tend to learn simple patterns first, then gradually memorize all samples [11, 12, 13]. These methods treat samples with small training loss as clean ones. During the training process, clean or confident instances are selected to update the model parameters. For instance, MentorNet [14] trains a Teacher-Net to provide curriculum, in the form of weights on the training samples, to select clean samples to guide the training of the Student-Net. Co-teaching [15] uses two networks to determine clean samples in their mini-batches separately and exchange the update information with the other network. Inspired from Co-teaching, JoCoR [16] also uses two networks, while they combine the ”agreement strategy” from semi-supervised learning into noise label tasks, which uses a joint loss to make their predictions agree. The instances with small joint loss are selected for the back-propagation. Experiments in JoCoR showed that it is more effective than co-teaching. However, not all clean examples can be selected by the network with the small loss strategy. When the noise rate is high, the selection would further decrease the number of effective training samples in each batch. Currently, there are relabelling methods learning network parameters and inferring the ground-truth labels simultaneously without any clean dataset, such as joint optimization [17] and PENCIL [18]. They use all data for learning so noisy examples can be an interference in this process.

In this paper, motivated by increasing the effective training samples, we proposed a framework that combines the noise correction with small loss selection methods to update the network parameters and noisy labels iteratively (Fig. 1). Each time we train relatively robust models with small loss examples, and correct the noisy labels which the current networks are confident on. In the training process, we are inspired by JoCoR to use ”agreement strategy” to train two networks with a joint loss. Our contributions are as follows.

(a) We proposed a robust learning framework for both network parameters and ground truth labels to handle noise label tasks. Our method is independent of the backbone network structure and does not need an auxiliary clean dataset. Training the network with small loss selection and updating the label of confident examples make the iterative learning process robust. To the best of our knowledge, it is the first method in this line. (b) We prove the effectiveness of training a stable model over current dataset before each label correction step. It performs better than conducting weights and label update at each iteration. (c) We conduct extensive experiments on both synthetic and real-world noisy datasets and our method achieves state-of-the-art accuracy.

[†] Authors contributed equally to this work. [*] Corresponding authors.

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

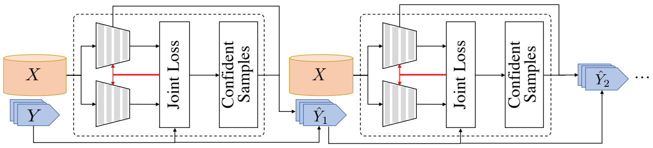


Fig. 1: The iterative network learning and label correction of proposed method. Two models are trained with a joint loss and confident samples are selected for relabelling.

2. PROPOSED APPROACH

2.1. Notation

For multi-class classification with C classes, we have a dataset D with N samples. $D = \{x_i, y_i\}_{i=1}^{N}$, where x_i is the ith example with its label as $y_i \in [1, ..., C]$. The set of examples and labels are denoted as X and Y. The neural network is denoted by $f(x, \theta)$, with the output of the final layer (C-class softmax layer): $\mathbf{p} = [p_1, ..., p_C]$. $\mathbf{p}$ is the predicted probability for each example x.

During network training in a standard classification task, a loss function $\mathcal{L}$ is used to measure the distance between $\mathbf{p}$ and the label y. The network parameters θ are learned by optimizing $\mathcal{L}$ by gradient descent methods. In our study, the given label Y is noisy, and our task is to jointly optimize θ and labels Y. The learned label set is denoted as $\hat{Y}$.

2.2. Joint Training with Small Loss Selection

According to the agreement maximization principles [19, 20, 16], different models would agree on labels of most examples and are unlikely to agree on incorrect labels. In the process of updating θ, two different classifiers are encouraged to make predictions closer to each other with a regularization term to reduce divergence. Specifically, joint training uses a joint loss to train two networks (same structure with different initialization) simultaneously. The loss function is composed of supervised learning loss and agreement loss as shown in Eq. 1,

$$\mathcal{L}_{joint} = (1 - \lambda) * \mathcal{L}_{sup} + \lambda * \mathcal{L}_{agr}, \tag{1}$$

where λ is a hyperparameter for linear combination and it decreases with the label noise rate.

The supervised learning loss $\mathcal{L}_{sup}$ is the sum of the classification loss of two networks, which is the Cross-Entropy between predictions and labels. The agreement loss is the term to reduce divergence between two classifiers which is the Jensen-Shannon divergence in Eq. 2, where $\mathcal{D}_{KL}(p_1||p_2)$ is the Kullback–Leibler divergence.

$$\mathcal{L}_{agr} = \mathcal{D}_{KL}(p_1||p_2) + \mathcal{D}_{KL}(p_2||p_1), \tag{2}$$

Small loss examples are selected to do the back-propagation in each iteration and a selection rate $R(t)$ is used to control the portion. $R(t)$ depends on the iteration t and its schedule is introduced in Section 2.4.

2.3. Learning with label correction

From the view of agreement maximization principles that different models are unlikely to agree on incorrect labels, the joint loss would be high with both high supervised learning loss and agreement loss over noisy labelled examples. Given this assumption, we obtain clean labels by updating labels in the direction to decrease Eq. 3.

$$\min_{\theta_1, \theta_2, Y} \mathcal{L}_{joint}(\theta_1, \theta_2, Y|X). \tag{3}$$

In the proposed learning framework, network parameters θ_1, θ_2 and class labels Y are alternatively updated: 1) Updating θ with fixed Y: With the joint loss in Eq. 1, we update θ_1 and θ_2 with small loss selection. 2) Updating Y with fixed θ_1, θ_2: After relatively robust models are trained on the current dataset, part of the noisy labels are corrected.

When updating the labels Y, we want to select the labels which are noisy and the networks are confident to update. The examples with small agreement loss are considered as confident examples so we select the subset $D_{correction}$ with the least agreement loss based on the label correction rate:

$$D_{confident} = argmin_{D':|D'| \geq C(k)|D|} \mathcal{L}_{agr}(D'), \tag{4}$$

where $C(k)$ is the label correction rate which changes with the k^{th} label correction. Among the confident examples, we take the noisy ones whose predicted class distributions have large difference with the current labels. Similarly, this subset D_{noisy} can be selected with the large supervised learning loss according to the label correction rate $C(k)$:

$$D_{noisy} = argmax_{D':|D'| \leq C(k)|D|} \mathcal{L}_{sup}(D'). \tag{5}$$

The subset for label correction $D_{correction}$ will be the intersection of D_{noisy} and $D_{confident}$:

$$D_{correction} = D_{confident} \cap D_{noisy}. \tag{6}$$

Within the selected examples, if two networks have the same predictions on one example, the current label is updated with the predicted class, otherwise it remains unchanged.

Instead of updating labels and parameters simultaneously in each iteration, we train two robust models on the current dataset before every label update step to make the correction more reliable. Since labels are not continuously updated, there can be a large amount of changes on the dataset after correction. We set a threshold $C_{restart}$ for label correction rate $C(k)$ as a trigger for restarting training. When $C(k) > C_{restart}$, a large portion of labels will be changed and we retrain two networks from scratch with the new labelled dataset.

We can use any deep neural network as the backbone network, and then equip it with the joint training and label correction to handle learning problems with noisy labels. After the networks have been fully trained, the label correction parts are not needed. The backbone networks alone can perform prediction for test examples and indicates ambiguous ones when two networks disagree.

2.4. Training implementation

The label correction rate $C(k) = \tau/2 * 1/k$ since as the number of correction time k increases, fewer examples are noise-labelled. The raw noise ratio τ decreases after label update according to the size of last $D_{correction}$:

$$\tau_k = \tau_{k-1} - \frac{|D_{correction}|}{|D|}. \tag{7}$$

The schedule of $R(t) = 1 - min\left\{\frac{t}{T_k}\tau, \tau\right\}$, where t is the iteration and $T_k = 10$ for CIFAR-10 and CIFAR-100, $T_k = 5$ for Clothing1M.

The training of our method is implemented through two stages: (1) Iterative optimization of network parameters θ_1, θ_2 and labels $\hat{Y}$. (2) Fine-tuning θ_1 and θ_2 with fixed labels $\hat{Y}$.

3. EXPERIMENTS

3.1. Datasets

Our method is demonstrated on synthetic datasets CIFAR-10, CIFAR-100, and noisy dataset Clothing1M [21], using the PyTorch framework. We corrupted synthetic datasets with symmetric and asymmetric noise manually. The asymmetric noise is to simulate that labellers may make mistakes only within very similar classes.

A label transition matrix Q is used to flip clean label y to noise label $\hat{y}$ [22, 23]. In the symmetric noise setup, label noise is uniformly distributed among all categories, and the label noise ratio is $\tau \in [0, 1]$. For each example, the noise-contaminated label has $1 - \tau$ probability to remain correct, but has τ probability to be flipped uniformly with other $c - 1$ labels. All datasets have the same symmetric noise setting. As for asymmetric noise, The noisy labels of CIFAR-10 were generated by mapping $truck \rightarrow automobile$, $deer \rightarrow horse$, $bird \rightarrow airplane$ and $cat \leftrightarrow dog$ [18] with probability τ. In CIFAR-100, 100 classes are grouped into 20 5-size super-classes and each class is flipped into the next circularly with noise ratio τ within each superclass [16].

Clothing1M is a large-scale dataset from 14 clothing classes with noisy labels. The estimated noise level is 40% [21]. we use the 1M images with noisy labels for training, the 14k and 10k clean data for validation and test, respectively. We resize the image to $256 * 256$ and crop the middle $224 * 224$ as input, then perform normalization. This dataset is seriously imbalanced among different classes and the label mistakes mostly happen between similar classes. PENCIL [18] randomly selected a small balanced subset to relieve the difficulty caused by imbalance.

3.2. Implementation details

We use a 7-layer CNN network architecture for CIFAR-10 and CIFAR-100 and 18-layer ResNet for Clothing1M, which are same in JoCoR for fair comparison.

For experiments on CIFAR-10 and CIFAR-100, Adam optimizer with momentum=0.9 is used and the batch size is 128. An initial learning rate of 0.001 is used in the first 250 epochs for iterative parameter and label learning. For fine-tuning the network with fixed labels, we run 50 epochs and linearly decay the learning rate from 0.001 to zero. The λ in Eq. 1 balances the supervised learning loss and agreement loss. We linearly decrease λ from 0.9 to 0.7 after each label update and keep 0.7 at the fine-tuning stage. The label update interval is 50 epochs. The label correction threshold $C_{retrain} = 5\%$.

In Clothing1M, the same Adam optimizer is used and the batch size is 64. We run 15 epochs in iterative optimization and 10 epochs for fine-tuning. The learning rate for the first 5 epochs is 8×10^{-4}, second 10 epochs is 5×10^{-4}, then for fine-tuning is 5×10^{-5}. λ is set to 0.85. $T_{update} = 5$ and the retraining threshold $C_{retrain} = 5\%$.

3.3. Comparison with baselines

We compare our method with following baseline algorithms on each dataset: F-correction [23], Co-teaching [15], Co-teaching+ [24], JoCoR [16], PENCIL [18], DivideMix (Clothing1M dataset) [25] and L_{DMI} [26]. The performance of standard deep networks training on noisy datasets is also used as a simple baseline. The performance of PENCIL, DivideMix and L_{DMI} is based on our implementation, and the others are quoted from [16].

CIFAR-10: The test accuracy and standard deviation compared with baselines over CIFAR-10 are shown in Table. 1. Our method performs the best in all four cases especially with symmetric-50% noise (+6.21). We can see as the noise ratio increased, the standard deviation increases which indicates the stability of models is affected by noise labels. Our method has a smaller standard deviation especially in the high noise rate. Our standard deviation is 0.89 in symmetry-80%, while that of the other top-3 methods co-teaching, JoCoR and PENCIL are 2.22, 3.06 and 1.86, respectively.

CIFAR-100: we compare our results with other baselines on CIFAR-100 in Table 2. CIFAR-100 has a similar dataset size to CIFAR-10 while with 10 times of classes, thus the classification is more challenging. Our method is still the overall accuracy winner. In symmetry-20% and symmetry-50%, Our method and JoCoR works significantly better than other methods and ours has an advantage of $+1.6 \sim 1.9$ over JoCoR. In the hardest case of symmetry-80%, JoCoR ties together with Co-teaching and ours has $+3.36$ test accuracy. The standard classifier, F-correction and PENCIL fail in this case with the accuracy below 5%. In terms of asymmetry-40% noise, Co-teaching+ performs better than other baselines with 33.62% accuracy whereas that of ours is 34.41%.

Clothing1M: We demonstrate our method on real-world noisy labels with Clothing1M dataset, which includes a lot of unknown structure (asymmetric) noise. The results are shown in Table. 3, where “Best” denotes the test accuracy of the epoch where the validation accuracy is optimal and “last” denotes the test accuracy of the last epoch. We use the complete noisy training set of Clothing1M instead of the small pseudo-balanced subset. Our method achieved SOTA performance

Table 1: Average test accuracy (%) and standard deviation on CIFAR-10 of 5 trials.

Flipping Rate	Standard	F-correction	Co-teaching	Co-teaching+	JoCoR	PENCIL	L_{DMI}	Ours
Symmetry-20%	69.18 ± 0.52	68.74 ± 0.20	78.23 ± 0.27	78.71 ± 0.34	85.73 ± 0.19	81.35 ± 0.32	80.52	**86.84 ± 0.18**
Symmetry-50%	42.71 ± 0.42	42.19 ± 0.60	71.30 ± 0.13	57.05 ± 0.54	79.41 ± 0.25	69.29 ±0.78	73.21	**85.62 ± 0.48**
Symmetry-80%	16.24 ± 0.39	15.88 ± 0.42	26.58 ± 2.22	24.19 ± 2.74	27.78 ± 3.06	25.30± 1.86	21.48	**29.01 ± 0.89**
Asymmetry-40%	69.43 ± 0.33	70.60 ± 0.40	73.78 ± 0.22	68.84 ± 0.20	76.36 ± 0.49	68.53±0.48	65.12	**76.83 ± 0.29**

Table 2: Average test accuracy (%) and standard deviation on CIFAR-100 of 5 trials.

Flipping Rate	Standard	F-correction	Co-teaching	Co-teaching+	JoCoR	PENCIL	L_{DMI}	Ours
Symmetry-20%	35.14 ± 0.44	37.95 ± 0.10	43.73 ± 0.16	49.27 ± 0.03	53.01 ± 0.04	43.61 ±0.23	48.95	**54.90 ± 0.07**
Symmetry-50%	16.97 ± 0.40	24.98 ± 1.82	34.96 ± 0.50	40.04 ± 0.70	43.49 ± 0.46	26.41 ±0.51	39.21	**45.12 ± 0.42**
Symmetry-80%	4.41 ± 0.14	2.10 ± 2.23	15.15 ± 0.46	13.44 ± 0.37	15.49 ± 0.98	3.65±0.77	10.65	**18.85 ± 0.59**
Asymmetry-40%	27.29 ± 0.25	25.94 ± 0.44	28.35 ± 0.25	33.62 ± 0.39	32.70 ± 0.35	27.32 ±0.42	31.34	**34.41 ± 0.72**

Table 3: Average accuracy (%) on Clothing1M test set.

Methods	Best	Last
Standard	67.22	64.68
F-correction [23]	68.93	65.36
Co-teaching [15]	69.21	68.51
Co-teaching+ [24]	59.32	58.79
JoCoR [16]	70.30	69.79
PENCIL [18]	69.48	68.48
L_{DMI}[26]	70.79	**70.02**
DivideMix [25]	68.24	67.19
Ours	**71.63**	69.97

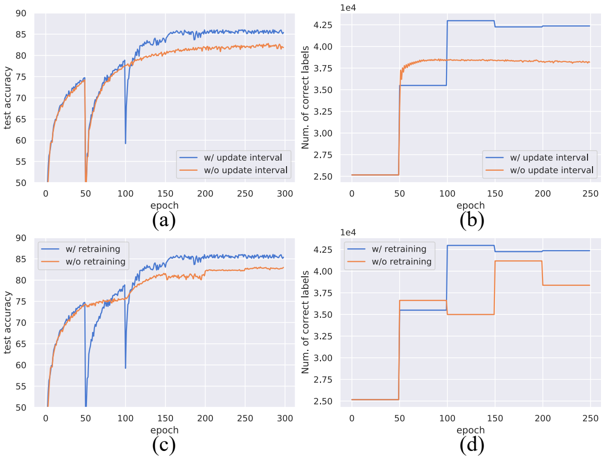
in "Best". In the last epoch, our performance is slightly decreased because of overfitting and 0.05% lower than L_{DMI}. The accuracy drop can be improved by stopping training early according to the validation set in practice.

3.4. Ablation Study

We conduct ablation study for analyzing the effect of label update interval and retraining. The experiments are set up on CIFAR-10 with Symmetry-50% noise.

Label Update Interval: Instead of referring clean labels in each iteration [17, 18], our label correction is intermittent. With label update intervals, two classifiers can be trained on the current dataset with small loss selection strategy and try to maximize the use of current data. The agreement maximization principle will be more reliable with these relatively robust models. In this experiment, we set the label update interval to 50 epochs and compared it with the continuous update in each iteration. Similar to Joint optimization [17], a pre-trained backbone is needed for continuous label update. We train two networks for 50 epochs before the label update both in continuous and intermittent methods.

Fig. 2 (a) and (b) show the test accuracy and number of correct labels versus epochs, respectively. In Fig. 2 (a), the large drop of the test accuracy indicates the trigger of retraining when the label correction rate is large. In continuous update, the label correction is smooth after the first correction. For fair comparison, we retrain networks with new labels after the first update. We can see the test accuracy with update intervals is better than continuous update, and the number of correct labels is significantly higher than the latter. It proves the effectiveness of using relatively robust models to do the label correction. Note that continuous correction has an average test accuracy of 82.1%, which still maintains an advantage over the other methods in Table. 1.



(a) (b) (c) (d)

Fig. 2: Comparison of ablation study over CIFAR-10. (a)(b): label update interval; (c)(d): retraining after correction.

Retraining After Correction Generally the network training with a fixed dataset is continuous for a large amount of epochs until it is stable. In Joint optimization and PENCIL [17, 18] with label update, their labels and weights learning are performed in every iteration and the dataset will not be changed dramatically after each label correction. In terms of our method, the labels may change a lot in every label update, because the correction ability of networks after the label correction interval can be improved much. In our experiment, retraining the models with the updated dataset has a better performance than continuous training as shown in Fig. 2 (c)(d). In Fig. 2 (d), the number of correct labels without retraining is less than retraining, and it drops a lot in the last update. Correspondingly, the test accuracy of the former is lower than the latter. However, the mean accuracy of 82.8% without retraining is still higher than that of other baselines.

4. CONCLUSION

We proposed a synergistic network learning and label correction methods to solve the noise label problem. Our method is independent of the backbone network thus it is easy to apply. We demonstrated our method with synthetic label noise on CIFAR-10 and CIFAR-100 and real-world large scale dataset Clothing1M with label noise. Our method is an overall winner compared with the baselines.

5. REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[3] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool, "Webvision database: Visual learning and understanding from web data," *ArXiv*, vol. abs/1708.02862, 2017.

[4] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten, "Exploring the limits of weakly supervised pretraining," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 181–196.

[5] Benoît Frénay and Michel Verleysen, "Classification in the presence of label noise: a survey," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 5, pp. 845–869, 2013.

[6] Shu Hu, Yiming Ying, Xin Wang, and Siwei Lyu, "Learning by minimizing the sum of ranked range," *NeurIPS*, vol. 33, 2020.

[7] Shu Hu, Lipeng Ke, Xin Wang, and Siwei Lyu, "Tkml-ap: Adversarial attacks to top-k multi-label learning," in *ICCV*, 2021, pp. 7649–7657.

[8] Shu Hu, Yiming Ying, Xin Wang, and Siwei Lyu, "Sum of ranked range loss for supervised learning," *arXiv preprint arXiv:2106.03300*, 2021.

[9] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[10] Tong Zhang and Bin Yu, "Boosting with early stopping: Convergence and consistency," *The Annals of Statistics*, vol. 33, no. 4, pp. 1538–1579, 2005.

[11] David Krueger, Nicolas Ballas, Stanislaw Jastrzebski, Devansh Arpit, Maxinder S Kanwal, Tegan Maharaj, Emmanuel Bengio, Asja Fischer, Aaron Courville, Simon Lacoste-Julien, et al., "A closer look at memorization in deep networks," in *International Conference on Machine Learning (ICML)*, 2017.

[12] Chan Lim, Sangwoo Han, and Jongwuk Lee, "Analyzing deep neural networks with noisy labels," in *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2020, pp. 571–574.

[13] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals, "Understanding deep learning (still) requires rethinking generalization," *Communications of the ACM*, vol. 64, no. 3, pp. 107–115, 2021.

[14] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei, "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *International Conference on Machine Learning*, 2018, pp. 2304–2313.

[15] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Advances in neural information processing systems*, 2018, pp. 8527–8537.

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

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

[18] Kun Yi and Jianxin Wu, "Probabilistic end-to-end noise correction for learning with noisy labels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7017–7025.

[19] Abhishek Kumar, Avishek Saha, and Hal Daume, "Co-regularization based semi-supervised domain adaptation," *Advances in neural information processing systems*, vol. 23, pp. 478–486, 2010.

[20] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu, "Deep mutual learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4320–4328.

[21] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang, "Learning from massive noisy labeled data for image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2691–2699.

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

[23] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1944–1952.

[24] X Yu, B Han, J Yao, G Niu, IW Tsang, and M Sugiyama, "How does disagreement help generalization against label corruption?," in *36th International Conference on Machine Learning, ICML 2019*, 2019.

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

[26] Yilun Xu, Peng Cao, Yuqing Kong, and Yizhou Wang, "L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise," in *Advances in neural information processing systems*, 2019, pp. 6225–6236.