Learning from Noisy Labels via Discrepant Collaborative Training

Yan Han[1,2], Soumava Kumar Roy[1,2], Lars Petersson[1,2], Mehrtash Harandi[2,3]
[1]The Australian National University, [2]DATA61-CSIRO, Australia, [3]Monash University
yan.han@anu.edu.au, soumava.kumarroy@anu.edu.au
Lars.Petersson@data61.csiro.au, mehrtash.harandi@monash.edu

Abstract

Noise is ubiquitous in the world around us. Difficulty in estimating the noise within a dataset makes learning from such a dataset a difficult and challenging task. In this paper, we propose a novel and effective learning framework in order to alleviate the adverse effects of noise within a dataset. Towards this aim, we modify a collaborative training framework to utilize discrepancy constraints between respective feature extractors enabling the learning of distinct, yet discriminative features, pacifying the adverse effects of noise. Empirical results of our proposed algorithm, Discrepant Collaborative Training (DCT), achieve competitive results against several current state-of-the-art algorithms across MNIST, CIFAR10 and CIFAR100, as well as large fine-grained image classification datasets such as CUBS-200-2011 and CARS196 for different levels of noise.

1. Introduction

Training a deep neural network on a large clean dataset, such as ImageNet [38], is not considered as a challenge any longer. However, it is expensive and time consuming to create large and clean datasets. If one is willing and able to accept noisy data in the training process, there is a vast amount of information easily accessible online. The issue of using such data is that the neural networks would easily overfit to the noise [42, 28, 27]. [44] shows that deep neural networks can easily fit random labels despite no apparent structure. This is true in particular for deeper and wider networks which may have the capacity to outrightly memorize the training data. As argued in [7], a noise rate of 20 can have a significantly detrimental effect, in some cases, halving the test accuracy.

Estimating the noise transition matrix directly is one of the popular methods to deal with a noisy dataset. For example, Goldberger et al.[12] added an additional softmax layer on top of the original softmax layer to model the noise transition matrix heuristically. However, when the number of classes increases, it becomes considerably more difficult to estimate the noise transition matrix. Additionally, the underlying assumption of the transition matrix is very strong. It assumes there is a fixed probability of an instance being corrupted into one of the other classes. An improvement to this could lead to focusing on identifying noisy samples by attempting to select a set of clean images instead of correcting the corrupt labels.

In order to move away from estimating the noise transition matrix, [19, 30, 35] proposed another direction which is to select clean instances out of a noisy dataset. Intuitively, an instance with a smaller loss has a greater probability of being correctly labeled. A neural network can then either use these examples to train or increase the proportion of these data.

In this paper, we propose a simple but effective learning paradigm called "Discrepant Collaborative Training". The proposed method allows us to make use of both clean and noisy examples, and improve the ability to select clean instances.

To identify clean samples from noisy ones, one can inspect the sample loss during training. Samples with a large loss value are more likely to be noisy and, hence, can be removed/weighted-down during the back propagation step.

The idea of Co-Training (Co-Tr) is to benefit from two (or more) networks which are complementary to one another and can help correct each other when they make mistakes. Naturally, the idea of Co-Tr suits the task of learning from a noisy dataset as well. The underlying assumption, here, is that the two networks are substantially different to each other so their loss signal is complementary. To achieve a discrepancy between the loss signals, one can choose to use two different network architectures. This, however, brings a new level of difficulty as, if the networks are not similar in structure, one might adapt faster with the result of the iterative procedure failing. Moreover, use of two different network architectures may lead to potential mismatch between their individual expressive powers, with a potential of one outperforming the other by a significant margin. As such, it may be more desirable in practice to use a specific structure for both networks and initialize them differently to achieve the necessary diversity [16, 19]. Unfortunately, even with this, there is no guarantee that the networks remain substantially different for the purpose of collaboration as required for the Co-Tr framework. In this work, we therefore propose a new method to ameliorate this shortcoming. A *max-discrepancy co-training framework* is proposed, where we achieve diversity by encouraging the networks to be statistically different. The notion of maximum mean discrepancy [17] is a simple yet powerful non-parametric criterion that measures the discrepancy of two distributions by mapping them to a Reproducing Kernel Hilbert Space (RKHS) [2].

Our main contributions are:

- A novel method to introduce diversity between networks in a co-training setting.
- Demonstrated improvement in identifying clean samples in a noisy dataset.
- Extensive experiments on different datasets and settings.
- Competitive or State-of-the-Art results with respect to comparable works on five different datasets, including two large fine-grained image recognition datasets (CUBS200-2011 [41] and CARS196 [22]).

2. Preliminaries

Notations: Throughout this paper, we use bold lower-case letters ($\boldsymbol{x}$) to show column vectors and bold upper-case letters (*e.g.*, $\boldsymbol{X}$) to show matrices.$[\cdot]_i$ is used to denote the i-th element of a vector and $\mathbf{I}_n$ shows the $n \times n$ identity matrix. The Frobenius norm of a matrix is shown by $\|\boldsymbol{X}\|_F = \sqrt{\mathrm{Tr}(\boldsymbol{X}^\top \boldsymbol{X})}$, with $\mathrm{Tr}(\cdot)$ indicating the matrix trace.

In this work, we make use of the Maximum Mean Discrepancy (MMD) [15] to measure the difference between two distributions $\mathcal{S}$ and $\mathcal{T}$. Let $\{\boldsymbol{X}_i^{\mathcal{S}}\}_{i=1}^n$ and $\{\boldsymbol{X}_i^{\mathcal{T}}\}_{i=1}^m$ denote i.i.d samples taken from $\mathcal{S}$ and $\mathcal{T}$, respectively.

An empirical estimate of MMD between $\mathcal{S}$ and $\mathcal{T}$ is obtained as

$$\mathrm{MMD}(\mathcal{S}, \mathcal{T}) = \left\| \frac{1}{n} \sum_{i=1}^{n} \Phi(\boldsymbol{X}_i^{\mathcal{S}}) - \frac{1}{m} \sum_{j=1}^{m} \Phi(\boldsymbol{X}_j^{\mathcal{T}}) \right\|_H^2 \quad (1)$$

Here, H denotes the induced Reproducing Kernel Hilbert space [1]) and $\|\cdot\|_H$ denotes its norm. $\mathrm{MMD}(\mathcal{S}, \mathcal{T})$ is a measure of overlap between $\mathcal{S}$ and $\mathcal{T}$ such that increase (or decrease) in overlap results in decrease (or increase) in $\mathrm{MMD}(\mathcal{S}, \mathcal{T})$. $\Phi(\boldsymbol{p})$ represents a functional mapping of the input $\boldsymbol{p}$ to a high-dimensional space. By the kernel trick, the form in Eqn. (1) can be written as;

$$\begin{aligned}\mathrm{MMD}(\mathcal{S}, \mathcal{T}) &= \frac{1}{n^2} \sum_{i}^{n} \sum_{i'}^{n} k(\boldsymbol{X}_i^{\mathcal{S}}, \boldsymbol{X}_{i'}^{\mathcal{S}}) \\ &- \frac{1}{nm} \sum_{i}^{n} \sum_{j}^{m} k(\boldsymbol{X}_i^{\mathcal{S}}, \boldsymbol{X}_j^{\mathcal{T}}) + \frac{1}{m^2} \sum_{j}^{m} \sum_{j'}^{m} k(\boldsymbol{X}_j^{\mathcal{T}}, \boldsymbol{X}_{j'}^{\mathcal{T}})\end{aligned} \quad (2)$$

In all our experiments, we have employed the Gaussian kernel

$$k(\boldsymbol{u}, \boldsymbol{v}) = \exp(-\frac{\|\boldsymbol{u} - \boldsymbol{v}\|^2}{\sigma}) \ . \quad (3)$$

The optimal value of σ used in Eqn. 3 is provided in the supplementary material.

3. Methodology

In this section, we present our proposed methodology *i.e.* Discrepant Colaborative Training (DCT). We formulate DCT with a cohort of two networks (see Fig. 1). The two sub-networks used in DCT are represented as f and g with its learnable parameters θ and $\widehat{\theta}$, respectively. We first briefly describe the selection strategy for choosing the clean labels, followed by the description of the discrepancy MMD module and the overall definition of the loss function.

3.1. Selection Strategy

Similar to [16, 19], we also employ the classification loss based selection strategy. More specifically, we obtain the classification loss of $\boldsymbol{X}_i$ for f and g as shown below:

$$\begin{aligned}\mathrm{L}_f(\boldsymbol{X}_i) &= -\log\left(\frac{\exp(\boldsymbol{z}_i^f)}{\sum_1^m \exp(\boldsymbol{z}_j^f)}\right) \\ \mathrm{L}_g(\boldsymbol{X}_i) &= -\log\left(\frac{\exp(\boldsymbol{z}_i^g)}{\sum_1^m \exp(\boldsymbol{z}_j^g)}\right) \quad ,\end{aligned} \quad (4)$$

where $\boldsymbol{z}_i^f = f_\theta(\boldsymbol{X}_i)$ and $\boldsymbol{z}_i^g = g_{\widehat{\theta}}(\boldsymbol{X}_i)$ [1]. Similar to [16], we select R examples for each f and g within a mini-batch of size N that produces the R lowest L_g and L_f respectively. Thereafter the loss to update θ_f is given as

$$\mathrm{L}_1^f = \sum_{i=1}^{R} \mathrm{L}_f(\boldsymbol{X}_i) \quad \forall \boldsymbol{X}_i \in \mathcal{D}_g \quad , \quad (5)$$

where $\mathcal{D}_g$ represents set of images that results in the R lowest L_g calculated in Eqn 4. Similarly, we calculate the loss to update θ_g as

$$\mathrm{L}_1^g = \sum_{i=1}^{R} \mathrm{L}_g(\boldsymbol{X}_i) \quad \forall \boldsymbol{X}_i \in \mathcal{D}_f \quad , \quad (6)$$

where $\mathcal{D}_f$ represents set of images that results in the R lowest L_f as calculated in Eqn 4.

[1] Usually it is the output of the softmax layer for each of f and g.

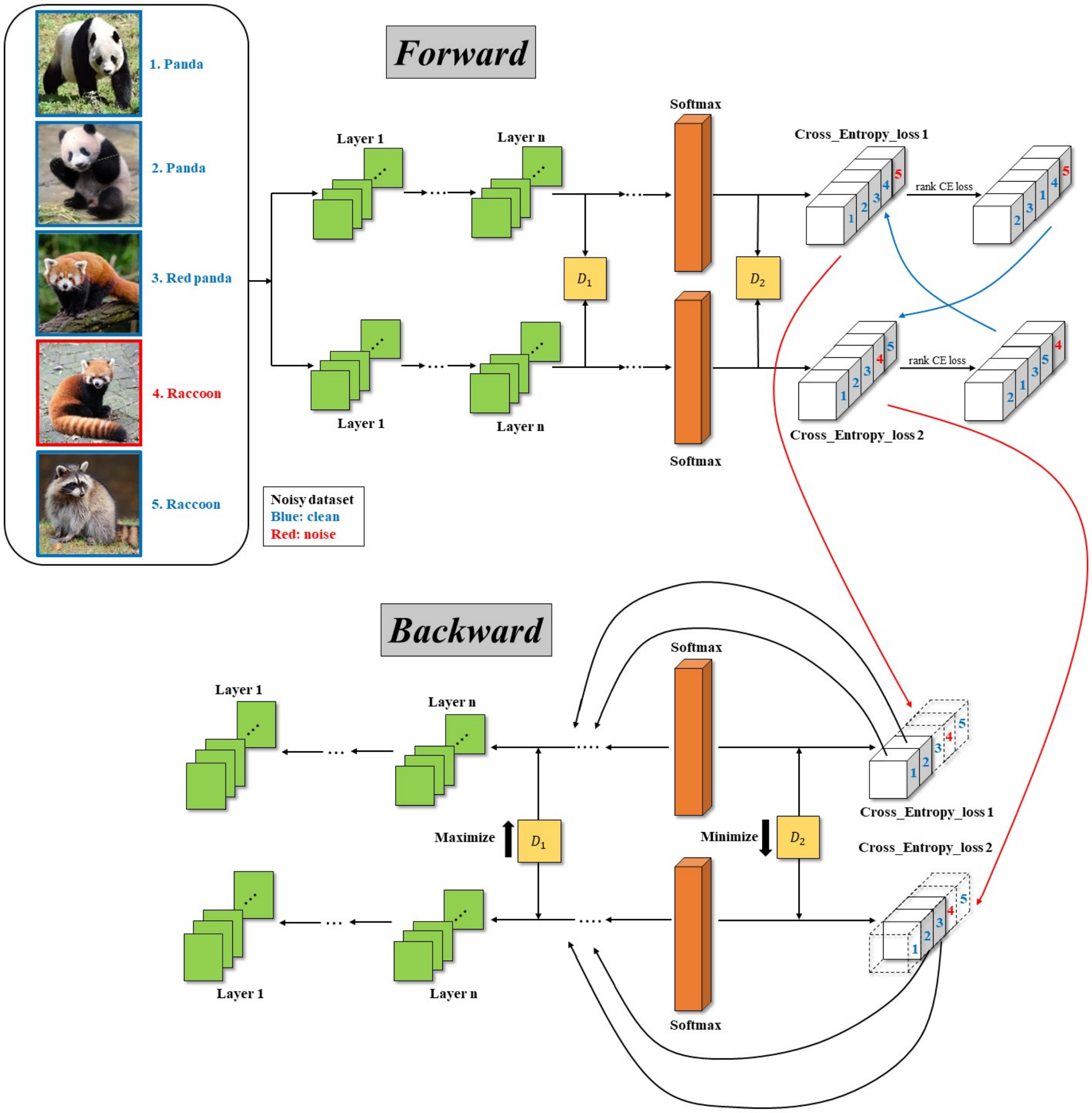


Figure 1. Forward (top) and backward (bottom) propagation of Discrepant Collaborative Training. **[FORWARD]** Five images are fed into sub-networks independently, four (blue) are correctly labeled and one (red) is corrupted with noise. The first discrepancy module is placed after *Layer n* between two networks. The second discrepancy module is placed after *softmax* layer. Then, the five images will be ranked according to its own cross entropy loss calculated by each network. Then the two networks exchange information of the ranking. **[BACKWARD]** According to the ranking information provided by its "peer" network, each network chooses only a few images with smaller loss value to update itself. We maximize the diversity of the first discrepancy module to learn diverse features in each of the network, while the diversity of the second module is minimized so as to learn the same class distributions.

3.2. Discrepancy Loss

In order for f and g to learn diverse features, we propose to explicitly insert a MMD module in-between them. The loss calculated is shown as:

$$\mathrm{L}_2 = \mathrm{MMD}(\boldsymbol{A}_i, \boldsymbol{B}_i) \tag{7}$$

where $\boldsymbol{A}_i = f_{\theta(1:l)}(\boldsymbol{X}_i)$ and $\boldsymbol{B}_i = g_{\widehat{\theta}(1:l)}(\boldsymbol{X}_i)$, l denotes the layer where the MMD module is used, and $\theta(1:l)$ and $\widehat{\theta}(1:l)$ represent the parameters of the two networks f and g till the layer l. It is to be noted that L_2 is calculated irrespective of the presence or absence of noisy labels within the mini-batch of the images.

3.3. Consistency Loss

Even though we want both f and g to learn diverse and distinct features, the final class probability distribution learnt by f and g should not be very different from each other. Thus we use another MMD module to explicitly reduce the discrepancy between the $\boldsymbol{z}_i^f$ and $\boldsymbol{z}_i^g$ for every $\boldsymbol{X}_i$ as shown below:

$$\mathrm{L}_3 = \mathrm{MMD}(\boldsymbol{z}_i^f, \boldsymbol{z}_i^g). \tag{8}$$

The final loss for DCT is given below:

$$\mathrm{Loss}_f = \mathrm{L}_1^f + \lambda_3 \mathrm{L}_3 - \lambda_2 \mathrm{L}_2 \tag{9}$$

$$\mathrm{Loss}_g = \mathrm{L}_1^g + \lambda_3 \mathrm{L}_3 - \lambda_2 \mathrm{L}_2 \tag{10}$$

λ_3 and λ_2 are the combination weights for the consistency and the diversity loss respectively. Eqn 9 and 10 are used to update f and g respectively using stochastic gradient optimizers. Algorithm 1 provides the pseudo code of our propose DTC algorithm.

4. Related Work

Co-Training : The seminal work was proposed by Blum *et al*. [4], where they successfully demonstrated the use of a Co-Tr framework to learn different insights in a standard Web page classification problem. In general, a Co-Tr algorithm attempts to learn two or more distinct and divergent feature extractors and have been successfully used across a multitude of tasks ranging from domain adaptation [6], image classification [34, 16], data segmentation [5] to tag-based image search [13] and many more. One such well-known algorithm is Learning to Teach [18]. Similar to Co-Tr, LT co-evolves a teacher and a student network(s) to learn discriminative features thanks to an inherent feedback sharing mechanism between the two of them. Similarly, model distillation algorithms [45, 29, 18] use an additional *mimicry* loss to align the final class-specific posterior distributions of every student network. Chen *et al*. [6] learns an Auto-encoder to benefit from using unlabeled data in the Co-Tr learning framework.

Discrepancy Measurement. Comparing and matching probability distributions between two different domains forms the fundamental building block for the design and development of several algorithms in the field of machine learning and computer vision [21, 14, 36]. Several divergences such as Kullback-Leibler (KL, [24]), Jensen-Shannon [31], *etc*. have been successfully integrated in learning similar/dissimilar distributions across various domains. However, these diversity learning algorithms do not take into account the geometry of the distributions, thereby failing to either disentangle or join the distributions [11]. Fortunately, several algorithms such as Maximum Mean Discrepancy (MMD) [17], Sinkhorn Divergence [11], Optimal Transport (OT) [3] *etc*. have been developed and successfully utilized to address the aforementioned drawback. MMD has been widely adopted as a discrepancy metric module across domain adaptation ([2]), unsupervised learning [43], generative models [9] and many more.

Learning from noisy datasets Learning from a clean dataset is not considered as a difficult task any longer [10, 37, 39]. Recently, there has observed a growing surge in the interest of studying the robustness of any machine learning algorithm against noisy labels. In this regard, MentorNet [19] trains an additional StudentNet network to select clean labels which is in-turn used to further guide the main training process. If a clean and unbiased validation set is not available, MentorNet will discover new data-driven sample-weight schemes from data which can be updated according to feedback from StudentNe. Ren *et al*. [35] follow a meta-learning paradigm and use a clean validation set to reweight the training samples. Importance weights for training samples which result in the decrease of the loss in a clean validation set are increased, while the weights of those that result in the increase of the loss are decreased during the training process. One of the major drawbacks of [35] is the calculation of the clean validation set based importance weights after every gradient update of the network, which increases the time complexity of the overall algorithm. On the other hand, Decoupling [30] trains two different sub-networks with the examples that are confusing to both of them during the course of training[2]. Similarly, Co-Tr [16] trains two different networks with one selecting the clean examples, *i.e*. the examples with lower classification loss, for the other in an intertwined fashion. However, without any explicit discrepancy module between the networks that enforce the features learnt to be distinct and different, the solution learnt by the two aforementioned algorithm is not optimal.

5. Empirical Evaluations

Dataset. We verify the effectiveness of our approach on five benchmark datasets: MNIST [26], CIFAR10 [23], CIFAR100 [23], CUB200-2011 [41] and CARS196 [22]. More details are listed in table 2.

Noise Type. We test our design on noisy-supervised image classification task. Since all datasets are clean, following [16, 33], we corrupt these datasets manually by the noise transition matrix $\boldsymbol{Q} \in \mathbb{R}^{K \times K}$, where $\boldsymbol{Q}_{ij} = Pr(\tilde{\boldsymbol{y}} = j | \boldsymbol{y} = i)$ gives the probability that noisy $\tilde{y}$ is flipped from clean y. In this paper, we test our methods on two differ-

[2] Both the networks produce different predictions with high confidence.

```
Data: $\theta$ and $\hat{\theta}$ (parameters of network $f$ and $g$), learning rate $\eta$, epoch number $T$, $l$ the layer number for the Diversity
    module.
Algorithm DCT
for $t = 1, \dots, T$ do
    Shuffle training set $\mathcal{D}$;
    Fetch mini-batch $\bar{\mathcal{D}}$ from $\mathcal{D}$;
    Obtain $\mathcal{D}_f = \arg\min_{\bar{\mathcal{D}}} l(f, \bar{\mathcal{D}})$;        sample R(T) instances with small cross entropy loss
    Obtain $\mathcal{D}_g = \arg\min_{\bar{\mathcal{D}}} l(g, \bar{\mathcal{D}})$;        sample R(T) instances with small cross entropy loss
    Update $\theta = \theta - \eta \left[\nabla l(\theta, \mathcal{D}_g) - \lambda_2 \nabla \mathrm{MMD}(f_{\theta(1:l)}(\bar{D}), g_{\hat{\theta}(1:l)}(\bar{D})) + \lambda_3 \nabla \mathrm{MMD}(f_\theta(\bar{D}), g_{\hat{\theta}}(\bar{D}))\right]$;
    Update $\hat{\theta} = \hat{\theta} - \eta \left[\nabla l(\hat{\theta}, \mathcal{D}_f) - \lambda_2 \nabla \mathrm{MMD}(f_{\theta(1:l)}(\bar{D}), g_{\hat{\theta}(1:l)}(\bar{D})) + \lambda_3 \nabla \mathrm{MMD}(f_\theta(\bar{D}), g_{\hat{\theta}}(\bar{D}))\right]$;
end
```

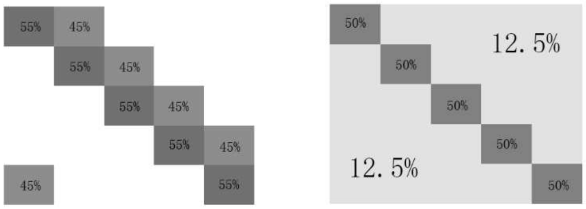
Algorithm 1: Discrepant Collaborative Training. Details of how to update R(T) is referred to [16].

Table 1. Structure of Network trained by MNIST, CIFAR10 and CIFAR100. The slope of all all LReLU functions in the network are set to 0.01. K denotes the number of training classes for the dataset used.

#Layer	Input Image
1	3×3 conv, 128 LReLU
2	3×3 conv, 128 LReLU
3	3×3 conv, 128 LReLU
	2×2 max-pool, stride 2 dropout, $p = 0.25$
4	3×3 conv, 256 LReLU
5	3×3 conv, 256 LReLU
6	3×3 conv, 256 LReLU
	2×2 max-pool, stride 2 dropout, $p = 0.25$
7	3×3 conv, 512 LReLU
8	3×3 conv, 256 LReLU
9	3×3 conv, 128 LReLU
10	avg-pool fc-layer 128 $\rightarrow K$ softmax

ent noise transitions matrix: (1)Symmetry flipping; (2) Pair flipping. Noise transition matrices are shown in figure 2(a) and 2(b). For easy understanding, we take noise rate $\epsilon = 45$ for pair flip and $\epsilon = 20, 50$ for symmetry flip.

In our experiments, we test three different noise conditions: (**1**) 45 pair flip noise; (**2**) 50 symmetry flip noise; (**3**) 20 symmetry flip noise. Since this paper mainly focuses on the value of noisy data and robustness of our proposed DCT, we choose high noise rate values *i.e.* 45 and 50. It is to be noted that, for pair flip noise, ϵ should be lower than 50, otherwise the neural network will need additional information to learn discriminative features. In order to verify the robustness of our DCT method under lower noise condition, we also test on 20 symmetry flip noise.



(a) Pair flip noise (b) Symmetry flip noise

Figure 2. Noise transition matrices [16]. We have shown for 5 classes as an example.

Network and optimizer. For experiments on MNIST, CIFAR10 and CIFAR100, we use the CNN architecture as shown in Table 1. We follow the settings of "Temporal Ensembling" ([25]) and "Virtual Adversarial Training" ([32]) which is the well-acknowledged standard test bed for weakly-supervised learning. Here, we use Adam [20] optimizer with momentum and initial learning rate set to 0.9 and 0.001 respectively. The batch size is fixed to 128 and we run DCT for 600 epochs. We choose [16] as our baseline and we follow their detailed settings. For experiments on CUB200-2011 and CARS196, we use Inception-V1 [40] architecture pretrained on Imagenet [8]. Here, we use RMSProp and Adam optimizer for CUB200-2011 and CARS196 dataset respectively.The initial learning rate for both the optimizers is set to 0.0001. The batch size is fixed to 32 and we report the results after 50 epochs of training.
Note: We report the optimal value of the hyper-parameters λ_2, λ_3 and σ used in DCT for all the datasets in the supplementary material.

Baselines. For MNIST, CIFAR10 and CIFAR100, we compare our results with the baseline methods: (i) F-correction [33], which corrects the prediction by using a noise transition matrix; (ii) Decoupling [30],which updates the parameters only using the samples where the two networks are not confident in their predictions; (iii) Mentor-

Table 2. Dataset Information

dataset	# of train images	# of test images	# of class	image size
MNIST	60,000	10,000	10	28×28
CIFAR10	50,000	10,000	10	32×32
CIFAR100	50,000	10,000	100	32×32
CUB200-2011	5,864	5,924	200	227×227
CARS196	8,054	8,131	196	227×227

Table 3. Comparison of our proposed DCT against several baseline algorithms. We report the average accuracy (%) after 5 runs for DCT.

noise	Dataset	F-correction [33]	Decoupling [30]	MentorNet [19]	Co-Teaching [16]	DCT
pairflip-45%	MNIST	0.24	58.03	80.88	87.63	**88.54**
symmetric-50%	MNIST	79.61	81.15	90.05	91.32	**94.21**
symmetric-20%	MNIST	**98.82**	95.70	96.70	97.25	98.54
pairflip-45%	CIFAR10	6.61	48.80	58.14	72.62	**72.91**
symmetric-50%	CIFAR10	59.83	51.49	71.10	74.02	**78.50**
symmetric-20%	CIFAR10	84.55	80.44	80.76	82.32	**85.41**
pairflip-45%	CIFAR100	1.60	26.05	31.60	34.811	**35.33**
symmetric-50%	CIFAR100	41.04	25.80	39.00	41.37	**42.11**
symmetric-20%	CIFAR100	**61.87**	44.52	52.13	54.23	56.11

Net [19], where an extra teacher network is pre-trained and used to filter out the noisy instances for its student network to learn robustly under noisy labels. (iv) Co-Teaching ([16]), which trains two identical sub-networks with one selecting the possible clean labels for the other. Results of the above baselines are reported from [16]. Furthermore, in order to verify the robustness of the performance of DCT method on the fine-grained image recognition datasets such as CUB200-2011 and CARS196; we consider two baselines model trained with the **(a)** conventional cross-entropy loss and **(b)** Co-Teaching algorithms.

5.1. Analysis on MNIST, CIFAR10 and CIFAR100

MNIST. As observed in Table 3, all the baseline methods obtain competitive results against each other for a low value of noise rate (*i.e.* symmetry 20%). Our DCT method obtains competitive 98.54% against the best method *i.e.* F-correction (which achieves 98.82%) accuracy on the test set. A drop in performance for F-correction and Decoupling baseline methods is observed when the noise rate is increased to 50%. On the other hand, MentorNet and Co-Teaching are pretty reboust in dealing with a higher noise rate. Our proposed DCT method outperforms all the baselines and obtains the state-of-the-art accuracy of 94.21%, outperforming the current best algorithm (*i.e.* Co-Teaching) by 2.89%. Further increase in the complexity of the nosie (*i.e.* pair-flip noise with $\epsilon = 45\%$) F-correction and Decoupling fail to classify images. Again, one can evidently observe that DCT outperforms all the baselines by a significant margin. More specifically, we outperform MentorNet by 7.66% in terms of accuracy on the test-set.

CIFAR10. From Table 3, it is observed that all the baseline methods obatin similar results in terms of accuracy on the test set for symmetry flip noise with $\epsilon = 20\%$. Unlike MNIST dataset, DCT outperforms all the baselines including F-correction by 0.86% for CIFAR10 dataset. On further increasing the complexity of the noise properties, it is easily observed that DCT is the best performing method against all the baselines, thereby validating the design choices of using two different MMD modules to learn distinct and discriminative features.

CIFAR100. It is observed from Table 3 that for symmetric noise with $\epsilon = 20\%$, DCT outperforms all the competitive baseline algorithms except F-correction. It is evident that F-correction is a reliable approach to learn from noisy labels for a lower value of noise rate. However, one can definitely observe that F-correction lacks robustness when the noise rate is increased. On the other hand, it is evident that DCT is more robust against the increase in noise percentages in comparison to the baseline methods; thereby reinforcing the choice of our algorithmic design.

5.2. Analysis on CUB200-2011 and CARS196

Figure 3(a) and 3(b) shows some exemplar images from CUB200-2011 and CARS196 dataset respectively.

CUB200-2011 The results are reported in Table 4. The baseline model trained with the conventional cross-entropy classification loss achieves a test accuracy of 63.78% for the symmetric noise with $\epsilon = 20\%$. An increase of 8.56%

Table 4. Comparison of our proposed DCT against several baseline algorithms for large fine-grained image recognition datasets in terms of accuracy on the test set (%).

noise	Dataset	Cross Entropy	Co-Teaching	DCT (ours)
symmetric-50%	CUB200-2011	40.80	54.64	**57.24**
symmetric-20%	CUB200-2011	63.78	72.34	**74.57**
symmetric-50%	CARS196	38.86	66.75	**67.80**
symmetric-20%	CARS196	71.76	86.00	**86.62**

Table 5. Study of the importance of using noisy samples. The average accuracy (%) is reported after 5 different runs of DCT.

noise	Dataset	Co-Teaching	DCT-clean	DCT
symmetric-50%	MNIST	91.32	92.92	**94.21**
symmetric-20%	MNIST	97.25	98.14	**98.54**
symmetric-50%	CIFAR10	74.02	76.81	**78.50**
symmetric-20%	CIFAR10	82.32	84.47	**85.41**
symmetric-50%	CIFAR100	41.37	41.55	**42.11**
symmetric-20%	CIFAR100	54.23	55.54	**56.11**

is observed over the cross-entropy baseline for the Co-Teaching algorithm. Morever, DCT outperforms all the baseline methods and achieves an accuracy of 74.57% for the same noise setting. Further increase in the ϵ to 50% evidently results in the decrease of the classification accuracy, however DCT still outperforms the rest by a significant margin. These results clearly demonstrate the effectiveness and robustness of DCT for large scale fine-grained image recognition dataset.

CARS196 As seen by the results obtained in Table 4, it is observed that our proposed DCT algorithm outperform both the Cross-Entropy baseline by a significant margin in terms of accuracy on the test set. It is however also observed that the performance gain over Co-Teaching is not substantial for CARS196 dataset in comparison to the performance gain obtained in CUBS200-2011. One plausible explanation that can be attributed to this trend is that CARS196 is a difficult dataset to train in comparison to CUBS200-2011.

According to the results shown in table 3 and 4, we verify the effectiveness and robustness of our DCT method, irrespective of the properties of the noise present in the datasets.

Note: One of the key aspects of the fine-grained image recognition datasets is that the inter-class variance is low and intra-class variance is high, and therefore are more vulnerable to noise. From the results obtained in Table 4, it is noted that the performance of Cross-Entropy baseline is substantially inferior against Co-Teaching and DCT. This observation clearly demands the need of two (or more) feature extractors in order to learn discriminative features in presence of noise with the fine-grained dataset.

6. Ablation Study

In this section, we perform extensive ablation study regarding the various design choices that have been considered for DCT.

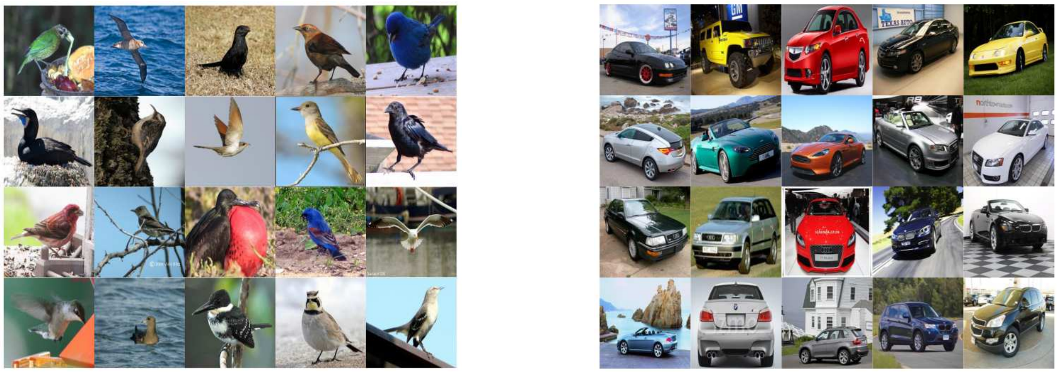
6.1. Importance of the selection strategy

As mentioned in § 3.2, L_2 in Eqn. 7 is not influenced by the selection strategy mentioned in § 3.1 as the entire mini-batch of is used to calculate L_2. In order to obtain more insights into the importance of the Diversity discrepancy module in the overall learning framework of DCT, we employ the selection strategy based on the ranking of the cross entropy loss to choose the samples for maximizing the divergence between the networks. In other words, we prune the noisy samples for the networks and attempt to increase the difference between the two networks [3]. We refer to this setting as *DCT-clean*. Similar to Eqn. (7), this operation is shown as follows:

$$\begin{aligned} &\mathrm{L}_2 = \mathrm{MMD}(\widehat{\boldsymbol{A}}_i, \widehat{\boldsymbol{B}}_i) \\ s.t.\ \ &\widehat{\boldsymbol{A}}_i = f_{\theta(1:l)}(\boldsymbol{X}_i) \qquad \forall \boldsymbol{X}_i \in \mathcal{D}_f \\ &\widehat{\boldsymbol{B}}_i = g_{\widehat{\theta}(1:l)}(\boldsymbol{X}_i) \qquad \forall \boldsymbol{X}_i \in \mathcal{D}_g \end{aligned} \tag{11}$$

The results are shown in Table 5. As observed, one can obtain better performance without such pruning of noisy labels, thereby successfully demonstrating the need of noisy labels to learn diverse features. One plausible explanation for this observation is that noisy labels perturb the overall decision boundary learnt by the two networks, thereby leading to optimal solution.

[3] It is to be noted that the clean labels of one network is chosen by the other network and vice-versa.



(a) CUBS200-2011

(b) CARS196

Figure 3. Exemplar images from the fine-grained image recognition datasets.

Table 6. Study of the importance of the two discrepancy modules used in DCT. w denote either of the feature extraction networks *i.e.*, f and g. (please refer to § 6.2 for more details.) The average accuracy ($\%$) is reported after 5 different runs of DCT.

noise	**Dataset**	Loss_w^D	Loss_w^C	**DCT**
symmetric-50%	MNIST	93.14	91.35	**94.21**
symmetric-20%	MNIST	97.89	97.04	**98.54**
symmetric-50%	CIFAR10	77.11	74.35	**78.50**
symmetric-20%	CIFAR10	84.08	82.42	**85.41**
symmetric-50%	CIFAR100	41.89	41.29	**42.11**
symmetric-20%	CIFAR100	55.48	54.13	**56.11**

Table 7. Study of position l for the Diversity loss module in the DCT framework. (please refer to § 6.3 for more details.) The average accuracy ($\%$) is reported after 5 different runs of DCT.

noise	**Dataset**	**DCT(7^{th})**	**DCT(5^{th})**
symmetric-50%	MNIST	93.27	94.21
symmetric-20%	MNIST	97.49	98.54
symmetric-50%	CIFAR10	77.69	78.50
symmetric-20%	CIFAR10	84.67	85.41
symmetric-50%	CIFAR100	41.90	42.11
symmetric-20%	CIFAR100	55.72	56.11

6.2. Importance of the Discrepancy Modules

In this section, we evaluate the importance of the two discrepancy modules, namely Diversity loss (*i.e.*, Eqn. (7)) and Consistency loss (*i.e.*, Eqn. (8)) in the DCT learning framework. We train the networks with the following loss functions

$$\text{Loss}_w^D = \text{L}_1^w + \text{L}_3 \quad \forall \ w = f, g \tag{12}$$

$$\text{Loss}_w^C = \text{L}_1^w - \text{L}_2 \quad \forall \ w = f, g \tag{13}$$

for the former and the later case respectively. The results are shown in Table 6. It is clearly observed that without use of diversity loss, the performance of DCT drops significantly. Surprisingly, with the removal of the consistency loss, no such drastic drop in the performance is observed. This clearly indicates the need of the Diversity loss module in DCT to learn distinct and discriminative features.

6.3. Importance of the Position of the Diversity Module

In this section, we study the effect of the position l in calculating the Diversity loss (please refer to Eqn. 7). In the original DCT algorithm, we have added the discrepancy module after the 5^{th} layer of the CNN. As an additional experiment, we also study the effect of fixing the discrepancy module after the 7^{th} layer. The results are shown in Table 7. As observed, fixing the discrepancy module after the 5^{th} layer leads to better results in comparison to the 7^{th}, although the difference in performance is not significant.

7. Conclusion

In this paper, we present a novel yet effective method (*i.e.* Co-Tr with Discrepancy) for training deep neural networks in the presence of noise. Specifically, we equip the Co-Tr framework with two different discrepancy modules, **(a)** Diversity and **(b)** Consistency. The former is aimed at enforcing discrepancy between the two feature extraction networks in the Co-Tr learning module, thereby learning distinct features; while the latter enforces the networks to learn similar class probability distributions. Our empirical evaluation across several datasets for different complex noise conditions clearly demonstrates the need of using such discrepancy modules in CTD. As an extension, the performance of different discrepancy modules other than MMD can be studied. Furthermore, noisy multi-label classification tasks is an area where similar approaches may prove successful.

References

[1] M. A. Alvarez, L. Rosasco, N. D. Lawrence, et al. Kernels for Vector-valued Functions: A Review. *Foundations and Trends® in Machine Learning*, 4(3):195–266, 2012. 2

[2] M. Baktashmotlagh, M. Harandi, and M. Salzmann. Distribution-Matching Embedding for Visual Domain Adaptation. *The Journal of Machine Learning Research*, 17(1):3760–3789, 2016. 2, 4

[3] R. Bhatia, T. Jain, and Y. Lim. On the Bures–Wasserstein Distance Between Positive Definite Matrices. *Expositiones Mathematicae*, 37(2):165–191, 2019. 4

[4] A. Blum and T. Mitchell. Combining Labeled and Unlabeled Data with Co-Training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. Citeseer, 1998. 4

[5] Y. Chai, V. Lempitsky, and A. Zisserman. Bicos: A Bi-level Co-Segmentation Method for Image Classification. In *2011 International Conference on Computer Vision*, pages 2579–2586. IEEE, 2011. 4

[6] S. Chen, G. Bortsova, A. G.-U. Juarez, G. van Tulder, and M. de Bruijne. Multi-Task Attention-Based Semi-Supervised Learning for Medical Image Segmentation. *arXiv preprint arXiv:1907.12303*, 2019. 4

[7] P. Craven and G. Wahba. Smoothing Noisy Data with Spline Functions. *Numerische mathematik*, 31(4):377–403, 1978. 1

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A Large-scale Hierarchical Image Database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[9] G. K. Dziugaite, D. M. Roy, and Z. Ghahramani. Training Generative Neural Networks via Maximum Mean Discrepancy Optimization. *arXiv preprint arXiv:1505.03906*, 2015. 4

[10] P. Fang, J. Zhou, S. K. Roy, L. Petersson, and M. Harandi. Bilinear attention networks for person retrieval. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8030–8039, 2019. 4

[11] J. Feydy, T. Séjourné, F.-X. Vialard, S.-I. Amari, A. Trouvé, and G. Peyré. Interpolating between Optimal Transport and MMD using Sinkhorn Divergences. *arXiv preprint arXiv:1810.08278*, 2018. 4

[12] J. Goldberger and E. Ben-Reuven. Training Deep Neural-Networks using a Noise Adaptation Layer. 2016. 1

[13] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik. A Multi-view Embedding Space for Modeling Internet Images, Tags, and Their Semantics. *International journal of computer vision*, 106(2):210–233, 2014. 4

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 4

[15] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A Kernel Two-sample Test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012. 2

[16] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama. Co-teaching: Robust Training of Deep Neural Networks with Extremely Noisy Labels. In *Advances in neural information processing systems*, pages 8527–8537, 2018. 2, 4, 5, 6

[17] J. Han, Y. Choi, and J. Lee. Characteristics of the Partially Reflected Terahertz Wave: Truncated Beam Propagation. *JOSA B*, 36(6):1551–1555, 2019. 2, 4

[18] G. Hinton, O. Vinyals, and J. Dean. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*, 2015. 4

[19] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei. Mentornet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels. *arXiv preprint arXiv:1712.05055*, 2017. 1, 2, 4, 6

[20] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[21] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013. 4

[22] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d Object Representations for Fine-Grained Categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013. 2, 4

[23] A. Krizhevsky, G. Hinton, et al. Learning Multiple Layers of Features from Tiny Images. Technical report, Citeseer, 2009. 4

[24] S. Kullback and R. A. Leibler. On Information and Sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951. 4

[25] S. Laine and T. Aila. Temporal Ensembling for Semi-Supervised Learning. *arXiv preprint arXiv:1610.02242*, 2016. 5

[26] Y. LeCun. The MNIST Database of Handwritten Digits. *http://yann. lecun. com/exdb/mnist/*, 1998. 4

[27] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L.-J. Li. Learning from Noisy Labels with Distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1910–1918, 2017. 1

[28] T. Liu and D. Tao. Classification with Noisy Labels by Importance Reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2015. 1

[29] D. Lopez-Paz, L. Bottou, B. Schölkopf, and V. Vapnik. Unifying Distillation and Privileged Information. *arXiv preprint arXiv:1511.03643*, 2015. 4

[30] E. Malach and S. Shalev-Shwartz. Decoupling "When to Update" from" How to Update". In *Advances in Neural Information Processing Systems*, pages 960–970, 2017. 1, 4, 5, 6

[31] C. D. Manning, C. D. Manning, and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT press, 1999. 4

[32] T. Miyato, A. M. Dai, and I. Goodfellow. Adversarial Training Methods for Semi-Supervised Text Classification. *arXiv preprint arXiv:1605.07725*, 2016. 5

[33] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu. Making Deep Neural Networks Robust to Label Noise: A Loss Correction Approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1944–1952, 2017. 4, 5, 6

[34] S. Qiao, W. Shen, Z. Zhang, B. Wang, and A. Yuille. Deep Co-Training for Semi-Supervised Image Recognition. In *The European Conference on Computer Vision (ECCV)*, September 2018. 4

[35] M. Ren, W. Zeng, B. Yang, and R. Urtasun. Learning to Reweight Examples for Robust Deep Learning. *arXiv preprint arXiv:1803.09050*, 2018. 1, 4

[36] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker Verification using Adapted Gaussian Mixture Models. *Digital signal processing*, 10(1-3):19–41, 2000. 4

[37] S. K. Roy, M. Harandi, R. Nock, and R. Hartley. Siamese networks: The tale of two manifolds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3046–3055, 2019. 4

[38] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 1

[39] C. Simon, P. Koniusz, and M. Harandi. Projective subspace networks for few-shot learning. 2018. 4

[40] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper with Convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 5

[41] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 2, 4

[42] X. Wu, R. He, Z. Sun, and T. Tan. A Light CNN for Deep Face Representation with Noisy Labels. *IEEE Transactions on Information Forensics and Security*, 13(11):2884–2896, 2018. 1

[43] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo. Mind the Class Weight Bias: Weighted Maximum Mean Discrepancy for Unsupervised Domain Adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 4

[44] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding Deep Learning Requires Rethinking Generalization. *arXiv preprint arXiv:1611.03530*, 2016. 1

[45] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu. Deep Mutual Learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 4