DISC: Learning from Noisy Labels via Dynamic Instance-Specific Selection and Correction

Yifan Li[1,2], Hu Han[1,2,3], Shiguang Shan[1,2,3], Xilin Chen[1,2]
[1]Key Laboratory of Intelligent Information Processing, Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing 100190, China
[2]University of the Chinese Academy of Sciences, Beijing 100049, China
[3]Peng Cheng Laboratory, Shenzhen 518055, China
{liyifan20g, hanhu, sgshan, xlchen}@ict.ac.cn

A. Additional Method Details

A.1. The Pseudo Code of DISC

The pseudo-code of DISC is shown in Algorithm 1. DISC optimizes a two-view (via weak and strong augmentation) based network, in which the prediction confidences from both views are used for calculating all losses. when calculating all the losses. DISC first warms up for T_0 epochs, then performs selection and correction in the rest epochs. To reduce confirmation bias, DISC utilizes the confidences from the previous epoch, which can also be regarded as a temporal ensembling of different iterations' results from the previous epoch.

A.2. The Setting of the Coefficient in Eq. (6), (7) and (12)

The custom method for calculating the losses is to average the values by the actual batch size. While in Eq. (6), (7), and (12), we choose to average the losses using the original batch size $\frac{1}{N}$ to resize the learning rate. Let θ denote the parameters of the model, then the updated parameters can be denoted by:

$$\Delta\theta = -\frac{\varepsilon}{N}\sum_{i=1}^{N}\nabla_\theta L(f_\theta(y_i;x_i),y_i), \tag{1}$$

where ε denotes the learning rate, and $f_\theta(x_i;y_i)$ denotes the model confidence for given label y_i. If the selected number is $(N_s < N)$, the updated parameters will be denoted by:

$$\Delta\theta = -\frac{k\varepsilon}{N_s}\sum_{i=1}^{N_s}\nabla_\theta L(f_\theta(y_i;x_i),y_i), \tag{2}$$

where k is the ratio to resize the learning rate. During the warm-up phase in which all the instances are utilized to calculate the loss. Thus, the learning rate should be proportional to the number of instances used to update the model parameters, i.e.:

$$\frac{k\varepsilon}{\varepsilon} = \frac{N_s}{N}. \tag{3}$$

Thus we have $k = \frac{N_s}{N}$. Replacing k in Eq. (2), and we can get Eq. (4). Eq. (4) actually reflects that the learning rate is proportional to the number used to update the model parameter, and with the number of selection or correction increasing, the learning rate increases too. Thus, we use $\frac{1}{N}$ to average the losses which can also be regarded as a weight for learning rate in Eq. (6), (7), (12).

$$\Delta\theta = -\frac{1}{N}\sum_{i=1}^{N_s}\nabla_\theta L(f_\theta(y_i;x_i),y_i), \tag{4}$$

Algorithm 1: DISC algorithm.

Input: Noisy set $\mathcal{N} = \{(x_i, y_i), \forall x_i \in \mathcal{X}, \forall y_i \in \mathcal{Y}, i = 1, ..., N\}$, start epoch T_0, total epochs T_{max}, model f_θ, λ, σ.

for $t = 1, ..., T_{max}$ **do**
  for $b = 1, ..., B$ **do**
    obtain batch images x_b and batch noisy labels y_b from data loader
    x_b^w, x_b^s = weakaug(x_b), strongaug(x_b)
    $p_w(c; x_b)$, $p_s(c; x_b)$ = $f_\theta(x_b^w)$, $f_\theta(x_b^s)$, $\forall c \in \mathcal{Y}$
    if $t \leq T_0$ **then**
      // warm-up for T_0 epochs
      $L = L_{\mathcal{C}}(p_w(c; x_b), p_s(c; x_b), y_b)$ // see formula (4)
    else
      obtain $\{x_b^c, y_b^c\}$, $\{x_b^h, y_b^h\}$, $\{x_b^m, \hat{\boldsymbol{y}}_b^m\}$ using $\mathcal{C}$, $\mathcal{H}$, $\mathcal{M}$ // obtain the subset of data batch
      $L = L_{\mathcal{C}}(p_w(c; x_b^c), p_s(c; x_b^c), y_b^c) + \lambda_h L_{\mathcal{H}}(p_w(c; x_b^h), p_s(c; x_b^h), y_b^h) + L_{\mathcal{M}}(p_w(c; x_b^m), p_s(c; x_b^m), \hat{\boldsymbol{y}}_b^m)$
      // see formula (5), (7), (10)
    end
    $\theta^{(b+1)}$ = SGD(L, θ^b) // update model parameters using optimizer
  end
  $\tau_w(t) = \lambda \tau_w(t) + (1 - \lambda) \max(p_w(t), dim = 1)$
  $\tau_s(t) = \lambda \tau_s(t) + (1 - \lambda) \max(p_s(t), dim = 1)$
  $\tau_{ws}(t) = 0.5\tau_w(t) + 0.5\tau_s(t)$
  $\tau'(t) = max(\tau_{ws}(t) + \sigma, 0.99)$
  /* Note that we use $\tau(t)$ or $\tau'(t)$ to obtain subsets at the end of each epoch, which means we do not select or correct instances in every iteration. Instead, we use the confidences of the previous epoch to get the subsets, which could be seen as a temporal ensemble of different iterations' results in the previous epoch. */
  $\mathcal{C} = \{x_i, y_i | p_w(y_i; x_i) > \tau_w(t)\} \cap \{x_i, y_i | p_s(y_i; x_i) > \tau_s(t)\}$
  $\mathcal{H} = \{x_i, y_i | p_w(y_i; x_i) > \tau_w(t)\} \cup \{x_i, y_i | p_s(y_i; x_i) > \tau_s(t)\} - \mathcal{C}$
  $p_{ws}(c; x_i) = 0.5p_w(c; x_i) + 0.5p_s(c; x_i), \forall c \in \mathcal{Y}, \forall x_i \in \mathcal{X}$
  $\mathcal{P} = \{x_i, \hat{y}_c = \arg\max_c p_{ws}(c; x_i) | \max_c p_{ws}(c; x_i) > \tau'(t), \forall c \in \mathcal{Y}\} - \{\mathcal{C} \cup \mathcal{H}\}$
  $\mathcal{M} = \{\mathcal{P} \cup \mathcal{C} \cup \mathcal{H}\}$
end

A.3. IDN Generation Algorithm

The pseudo-code of instance-depend noise (IDN) generation process is provided in Algorithm 2, which follows the method in [12].

B. Additional Experiment Details and Results

B.1. Additional Training Details

We provide the hyper-parameters for DISC in Table 1. From Table 1 we can see that our method only has four main hyper-parameters ($T_0, \lambda, \lambda_h, \sigma$) apart from the hyper-parameters about the optimizer. Furthermore, most of these hyper-parameters don't have to be modified in most cases, which verifies the robustness of our method DISC to hyper-parameters.

For the situation that noise ratio is extremely large (e.g., inst. 0.6), we use a relative small λ and λ_h and a relative large σ. Since λ controls the delaying degree and threshold stability, it's suitable to use a smaller λ for the situation when label noises are heavy (a smaller λ refer to a larger threshold). Similarly, if label noises are heavy, we should use a larger threshold to correct the noisy labels to reduce the contamination of noisy labels, and thus we use a larger σ in this situation. Moreover, since the label noises in hard set will also increase as the degree of label noises increases, we decrease the weight of $L_{\mathcal{H}}$ by reducing λ_h.

Algorithm 2: IDN generation algorithm.

Input: Noisy set $\mathcal{N} = \{(x_i, y_i), \forall x_i \in \mathcal{R}^{S\times 1}$, noise ratio ρ.

Sample instance noise rate q_i from the truncated normal distribution $\mathcal{N}(\rho, 0.1^2, [0, 1])$

Sample $W \in \mathcal{R}^{K\times S\times K}$ from the standard normal distribution $\mathcal{N}(0, 1^2)$ // K and S indicate the number of classes and the dimension of instance features

for $i = 1, 2, ..., N$ **do**

  $p = x_i^T \times W_{y_i}$ // generate instance-dependent flip rates

  $p_{y_i} = -\infty$ // control the diagonal entry of the instance-dependent transition matrix

  $p = q_i \times softmax(p)$ // make the sum of the off-diagonal entries of the y_i-th row to be q_i

  $p_{y_i} = 1 - q_i$

  Randomly choose a label from the label space according to the possibilities p as noisy label $\tilde{y}_i$

end

Output: Noisy instances $\{(x_i, \tilde{y}_i)\}_{i=1}^n$.

Table 1. The hyper-parameters for DISC on different benchmarks. T_{max} and T_0 represent the total epochs and warm-up epochs, respectively. λ is the layback ratio that controls the delaying degree and threshold stability. λ_h and σ indicate the weight of $L_{\mathcal{H}}$ and the offset of $\tau'(t)$, respectively. lr, bs and wd are the abbreviations of the learning rate, batch size, and weight decay, respectively.

Dataset	Backbone	T_{max}	T_0	λ	λ_h	σ	lr	bs	wd	learning rate schedule
CIFAR-10 (Inst. 0.2)	ResNet-34 (scratch)	200	15	0.99	1	0.3	0.1	128	1e-3	divide 10 at epoch 80, 160
CIFAR-10 (Inst. 0.4)	ResNet-34 (scratch)	200	15	0.99	1	0.3	0.1	128	1e-3	divide 10 at epoch 80, 160
CIFAR-10 (Inst. 0.6)	ResNet-34 (scratch)	200	15	0.95	0.2	0.5	0.1	128	1e-3	divide 10 at epoch 80, 160
CIFAR-100 (Inst. 0.2)	ResNet-34 (scratch)	200	15	0.99	1	0.3	0.1	128	1e-3	divide 10 at epoch 80, 160
CIFAR-100 (Inst. 0.4)	ResNet-34 (scratch)	200	15	0.99	1	0.3	0.1	128	1e-3	divide 10 at epoch 80, 160
CIFAR-100 (Inst. 0.6)	ResNet-34 (scratch)	200	15	0.95	0.2	0.5	0.1	128	1e-3	divide 10 at epoch 80, 160
Tiny-ImageNet (sym. 0)	PresNet-18 (scratch)	200	15	0.99	1	0.3	0.1	128	1e-3	divide 10 at epoch 80, 160
Tiny-ImageNet (sym. 0.2)	PresNet-18 (scratch)	200	15	0.99	1	0.3	0.1	128	1e-3	divide 10 at epoch 80, 160
Tiny-ImageNet (sym. 0.5)	PresNet-18 (scratch)	200	15	0.99	1	0.3	0.1	128	1e-3	divide 10 at epoch 80, 160
Tiny-ImageNet (asym. 0.45)	PresNet-18 (scratch)	200	15	0.95	0.2	0.3	0.1	128	1e-3	divide 10 at epoch 80, 160
Animal-10N	Vgg-19 (scratch)	120	10	0.99	1	0.3	0.05	64	5e-4	divide 10 at epoch 50, 80
Food-101	ResNet-50 (ImageNet)	100	10	0.99	1	0.3	0.01	32	5e-4	divide 10 at epoch 50, 80
WebVision	InceptionResNetV2 (scratch)	100	15	0.99	1	0.3	0.2	32	5e-4	divide 10 at epoch 50, 80
Clothing1M	ResNet-50 (ImageNet)	100	30	0.95	1	0.3	0.01	32	5e-4	divide 10 at epoch 50, 80

B.2. Additional Results on CIFAR-10/100

Besides the results of symmetric (sym.) and asymmetric (asym.) noise on the Mini-ImageNet database, we also provide the results of DISC on CIFAR-10/100 under sym. and asym. noise situations. There are two types of sym. noises according to whether the flipped labels contain the correct labels or not. We provide the results for both cases, i.e., with Table 2 showing the results of flipped labels without correct labels, and Table 3 showing the results of flipped labels containing correct labels. Three different noise rates are adopted in both two sym. noise types, i.e., $\rho \in \{20\%, 50\%, 80\%\}$. We provide the results on CIFAR-10/100 with 40% asym. noises following methods [3, 4, 11, 18] in Table 3.

For a fair comparison, we reproduced the results of all the baseline methods using their open-sourced codes and with the same backbone (PresNet-18) and training epochs (200). For all the methods, we use an SGD optimizer with a momentum of 0.9, a weight decay of 0.001, and a batch size of 128 for training. The initial learning rate is set as 0.1, decaying by a factor of 0.1 in epochs 80 and 160.

Results in Tables 2 and 3 show that our method DISC outperforms the state-of-the-art (SOTA) methods in most noise cases, such as DivideMix [4], ELR+ [7], Co-learning [9], RRL [5] and GJS [2]. In particular, while RRL performs well under the low rate of sym. noise situation, its robustness under high rate of sym. noise is not good. Compared with DivideMix and ELR+ which use two networks to perform LNL, our DISC uses a single network, but still performs better. Overall, our DISC shows strong capability and robustness in learning from data with a high percentage of sym. and asym. noise. DISC can handle the label noises with a large ratio, which shows stronger robustness and generalization ability.

Table 2. Test accuracies (%) and standard deviations over the last 10 epochs of different methods on CIFAR-10 and CIFAR-100 with different percentages of sym. and asym. noise, in which correct labels are used for noise generation. The results of all the baseline methods are reproduced using the open-sourced code with 200-epoch training. For a fair comparison, all the models are trained with PresNet-18 as a backbone and run three times with random seeds.

Dataset	CIFAR-10				CIFAR-100			
Noise type	Sym 20%	Sym 50%	Sym 80%	Asym 40%	Sym 20%	Sym 50%	Sym 80%	Asym 40%
CE	83.31 ± 0.09	56.41 ± 0.32	18.52 ± 0.16	77.06 ± 0.26	55.17 ± 0.12	32.40 ± 0.16	7.70 ± 0.06	40.63 ± 0.26
Mixup [17]	90.17 ± 0.12	70.94 ± 0.26	47.15 ± 0.37	82.68 ± 0.38	63.65 ± 0.29	40.94 ± 0.39	14.11 ± 0.31	46.83 ± 0.24
Decoupling [8]	85.40 ± 0.12	68.57 ± 0.34	41.08 ± 0.24	78.67 ± 0.81	52.75 ± 0.11	27.59 ± 0.16	7.38 ± 0.09	39.12 ± 0.08
Co-teaching [3]	87.95 ± 0.07	48.60 ± 0.19	17.48 ± 0.11	71.14 ± 0.32	56.03 ± 0.14	25.33 ± 0.08	5.18 ± 0.05	38.80 ± 0.08
JointOptim [10]	91.34 ± 0.40	89.28 ± 0.74	59.67 ± 0.27	90.63 ± 0.39	58.50 ± 0.47	53.58 ± 0.63	24.62 ± 0.50	61.17 ± 0.39
Co-teaching+ [15]	87.20 ± 0.08	54.24 ± 0.23	22.26 ± 0.55	79.91 ± 0.46	51.24 ± 0.23	25.07 ± 0.18	9.50 ± 0.08	35.66 ± 0.10
GCE [18]	90.05 ± 0.10	79.40 ± 0.20	20.67 ± 0.11	74.73 ± 0.39	59.92 ± 0.15	50.02 ± 0.12	18.53 ± 0.09	39.38 ± 0.17
PENCIL [14]	88.02 ± 0.90	70.44 ± 1.09	23.20 ± 0.81	76.91 ± 0.26	55.17 ± 0.12	32.40 ± 0.16	7.70 ± 0.06	40.63 ± 0.26
JoCoR [11]	89.46 ± 0.04	54.33 ± 0.12	18.31 ± 0.11	70.98 ± 0.21	54.70 ± 0.08	26.45 ± 0.13	5.50 ± 0.05	37.93 ± 0.09
DivideMix [4]	95.58 ± 0.06	94.73 ± 0.04	81.35 ± 0.36	91.69 ± 0.13	76.16 ± 0.11	72.84 ± 0.12	51.25 ± 0.07	55.56 ± 0.53
ELR [7]	90.35 ± 0.04	87.40 ± 3.86	55.69 ± 1.00	89.77 ± 0.12	72.88 ± 0.08	67.06 ± 0.09	28.40 ± 0.06	69.56 ± 0.07
ELR+ [7]	95.27 ± 0.11	94.41 ± 0.11	81.86 ± 0.23	91.38 ± 0.50	76.94 ± 0.18	73.01 ± 0.17	**58.01 ± 0.17**	74.39 ± 0.17
Co-learning [9]	92.14 ± 0.09	77.99 ± 0.65	43.80 ± 0.76	82.70 ± 0.40	69.93 ± 0.14	59.56 ± 0.18	41.77 ± 0.32	51.50 ± 0.24
RRL [5]	95.63 ± 0.06	93.96 ± 0.05	51.26 ± 0.20	92.44 ± 0.06	**79.22 ± 0.12**	74.81 ± 0.15	N/A	52.58 ± 0.22
GJS [2]	94.90 ± 0.13	89.22 ± 0.33	18.78 ± 0.18	85.95 ± 1.47	78.26 ± 0.17	21.58 ± 0.06	17.54 ± 0.22	58.73 ± 0.49
DISC (ours)	**96.10± 0.05**	**95.14 ± 0.07**	**84.69 ± 0.17**	**94.64 ± 0.04**	78.75 ± 0.13	**75.21 ± 0.15**	57.61 ± 0.29	**76.50 ± 0.15**

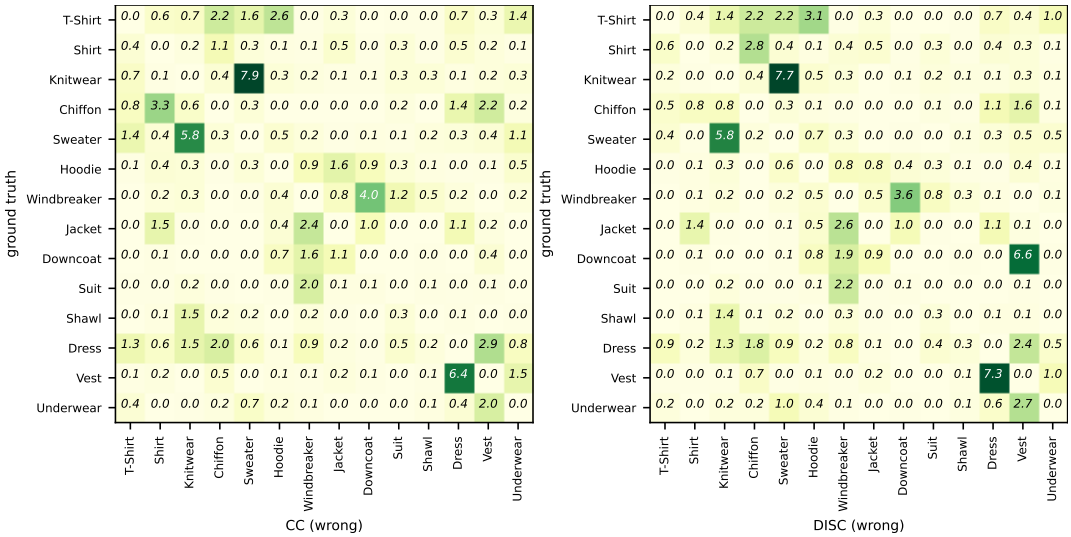
Table 3. Test accuracies (%) and standard deviations over the last 10 epochs of different methods on benchmark CIFAR-10 and CIFAR-100 under symmetric noise (containing clean labels) and percents. The results of all the baseline methods are reproduced using the open-sourced code with 200-epoch training. For fair comparison, all the models are trained with PresNet-18 as a backbone and run three times with random seeds.

Dataset	CIFAR-10			CIFAR-100		
Noise type	Sym 20%	Sym 50%	Sym 80%	Sym 20%	Sym 50%	Sym 80%
Cross-Entropy	84.61 ± 0.10	62.35 ± 0.30	27.28 ± 0.30	56.10 ± 0.12	33.60 ± 0.20	8.18 ± 0.09
Mixup [17]	90.65 ± 0.15	74.60 ± 0.52	50.93 ± 0.79	64.06 ± 0.31	42.86 ± 0.50	14.79 ± 0.38
Decoupling [8]	86.36 ± 0.12	72.92 ± 0.23	48.44 ± 0.55	53.28 ± 0.11	28.00 ± 0.10	7.89 ± 0.06
Co-teaching [3]	89.57 ± 0.07	62.53 ± 0.19	28.51 ± 0.06	62.31 ± 0.13	34.90 ± 0.09	7.11 ± 0.05
Co-teaching+ [15]	88.05 ± 0.04	61.81 ± 0.19	22.26 ± 0.55	54.53 ± 0.11	27.59 ± 0.10	8.37 ± 0.06
GCE [18]	90.92 ± 0.06	83.00 ± 0.12	33.06 ± 0.24	61.23 ± 0.15	48.33 ± 0.13	18.51 ± 0.14
JointOptim [10]	91.20 ± 0.54	89.57 ± 0.55	81.79 ± 0.34	81.79 ± 0.34	58.60 ± 0.84	53.72 ± 0.83
PENCIL [14]	88.24 ± 0.61	73.41 ± 1.52	36.00 ± 0.69	57.35 ± 0.99	11.42 ± 3.22	5.36 ± 1.17
JoCoR [11]	90.99 ± 0.05	76.11 ± 0.05	27.36 ± 0.09	64.24 ± 0.10	36.78 ± 0.08	7.65 ± 0.04
DivideMix [4]	95.77 ± 0.08	94.57 ± 0.10	92.65 ± 0.07	76.51 ± 0.13	73.33 ± 0.13	52.39 ± 0.17
ELR [7]	89.76 ± 3.32	88.54 ± 2.74	77.91 ± 4.08	73.48 ± 0.08	67.26 ± 0.08	29.41 ± 0.09
ELR+ [7]	95.22 ± 0.11	94.56 ± 0.16	91.76 ± 0.12	76.94 ± 0.18	73.01 ± 0.18	58.28 ± 0.18
Co-learning [9]	92.84 ± 0.09	82.40 ± 0.28	52.71 ± 0.72	69.82 ± 0.26	60.44 ± 0.21	40.41 ± 0.47
RRL [5]	**96.32 ± 0.06**	94.65 ± 0.08	82.03 ± 0.20	**78.93 ± 0.14**	74.96 ± 0.14	N/A
GJS [2]	95.16 ± 0.12	91.65 ± 0.30	71.46 ± 1.38	78.42 ± 0.17	21.98 ± 0.61	22.35 ± 0.21
DISC (ours)	96.25± 0.04	**95.40 ± 0.05**	**92.91 ± 0.11**	78.63 ± 0.09	**76.28 ± 0.14**	**59.34 ± 0.16**

B.3. Confusing Labels Observed on Clothing1M's Test Set

Clothing1M [13] is a well-known dataset for LNL due to its large size. However, similar to the label errors observed in the ImageNet [1, 16], we notice there are a large number of label errors of Clothing1M. Such kind of label errors may lead to biased testing accuracy of individual methods. Here, we show such kind of bias on Clothing1M using our DISC, and a SOTA method CC [19], which achieve 73.77% and 75.81% (using the original random seed) test accuracy, respectively on the test set of Clothing1M.

We analyze the failed cases of two methods using a confusion matrix (see Fig. 1 (a) and (b)), and visualize some of the failed cases in Fig. 2, in which the 'ground truth' labels (or test set labels) are shown above the images, and the predicted labels are shown below. From Fig. 1 (a) and (b) we can see that the confusing classes by the two methods are very similar. In general, we can summarize the failed cases by two methods into three cases, (i) the semantic overlapping labels, (ii) incorrect labels, and (iii) images containing contents of multiple classes. For the semantic overlapping case we mean the classes may



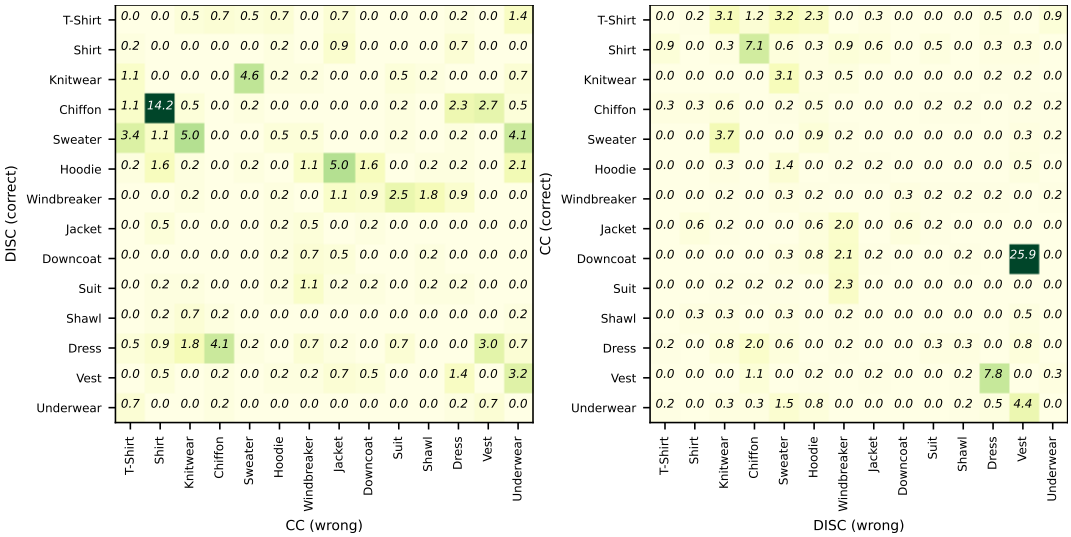
(a) (b)

Figure 1. The statistics of the incorrect prediction by CC (a) and DISC (b), respectively.

Figure 2. The noisy labels in test set on Clothing1M. The ground-truth labels (red) are above the images, and the prediction labels (black) are shown below. (a) Semantic overlapping labels. (b) Incorrect labels. (c) Multi-labels.

be all correct when referring to certain clothes since these classes are semantically closed (see Fig. 2 (a)), for instance, the 'knitwear' and 'sweater', 'vest' and 'dress', 'downcoat' and 'vest', 'chiffon' and 'vest', etc. For incorrect label cases, we mean the given labels in the test set are completely incorrect. For example, the 'dress' is mislabeled as a 'jacket', the 'hoodie' is mislabeled as a 'downcoat', etc. For the multi-label case, an image may contain multiple instances of different classes (see Fig. 2 (c)). For example, the first image in Fig. 2 (c) contains both 'vest' and 'dress', but only a single label 'dress' is given in the dataset.

We also provide quantitative analysis about the results by CC and DISC. We use a confusion matrix manner to indicate



(a) (b)

Figure 3. (a) statistics of the incorrect prediction by DISC and correct prediction by CC, and (b) vice versa.

the situation that the one method's predictions are correct while the other's predictions are wrong (see Fig. 3 (a) and (b)). From Fig. 3 (a) and (b), we can see that the easily confusing classes are different between the two methods. For instance, more than a quarter of 'downcoat' images are classified as 'vest' by DISC. However, as we check these cases, these images are almost 'vest downcoat', which means it's acceptable to classify them as either 'downcoat' or 'vest' (see the second image in Fig. 2 (a)). Similarly, around 14.2 % of 'chiffon' images are classified as 'shirt' by our DISC. After manually checking these cases, a number of 'chiffon shirt' images commonly appear together with women's wear. Based on our estimation, around 60% images for which CC is correct while DISC is wrong to have confusing labels. And there are about 50% images containing confusing labels for the opposite cases. Based on the above analyses, we argue that confusing labels in the test set of Clothing1M may lead to biased results, while it can serve as a reference for comparing different LNL methods.

B.4. The Visualization of Features

We provide the visualization result of high-dimension features extracted by DNN using t-SNE [6] under 40% asym. label noise on the CIFAR-10 benchmark, which is shown in Fig. 4. We can see that as training goes on, the noisy feature cluster gets denser and different class clusters are more dispersed. We attribute this phenomenon to the memorization strength, which increases along with the training.

C. Negative Social Impacts

Since all the benchmarks we used are public without face images inside, there are not known IRB problems.

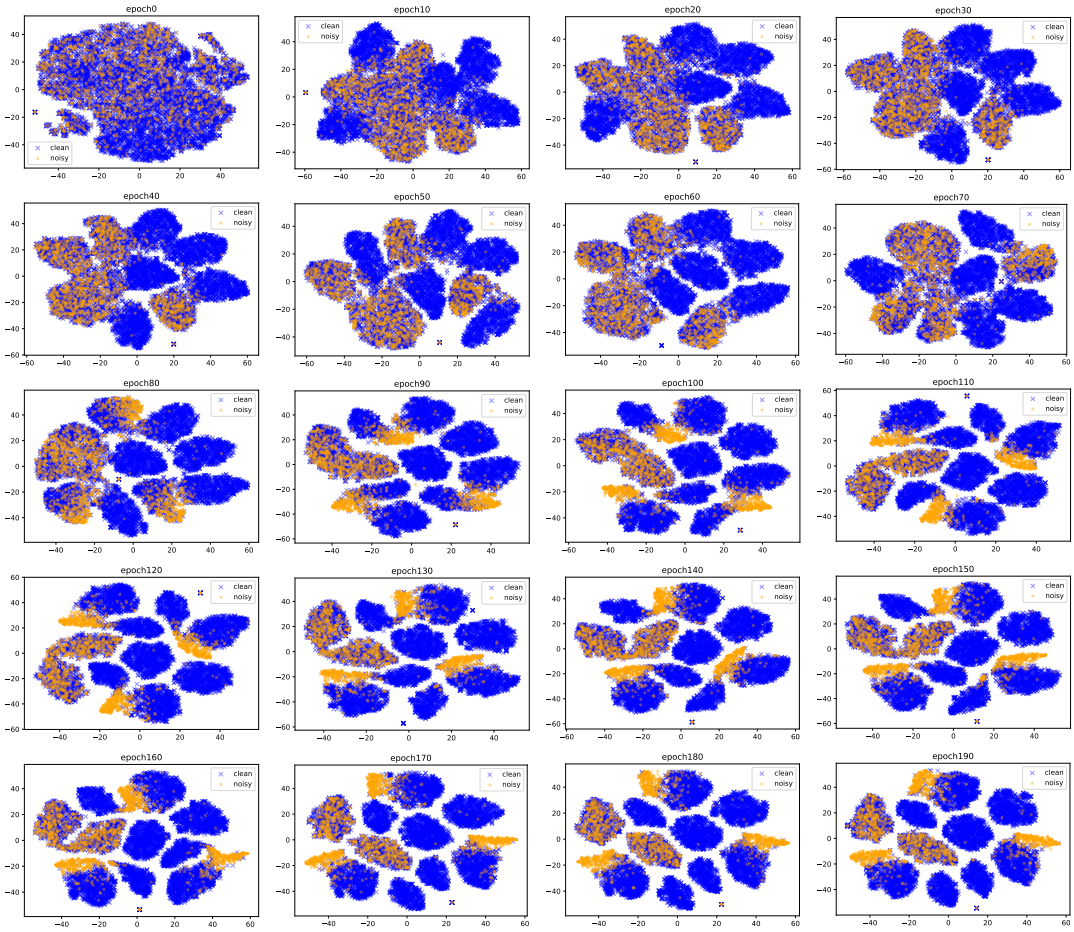


Figure 4. The visualization of the features using t-SNE under 40% asym. label noise on CIFAR-10 benchmark ranging from epoch 0 to epoch 190.

References

[1] Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? In *arXiv preprint arXiv:2006.07159*, 2020. 4

[2] Erik Englesson and Hossein Azizpour. Generalized jensen-shannon divergence loss for learning with noisy labels. In *Proc. NeurIPS*, volume 34, 2021. 3, 4

[3] 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 *Proc. NeurIPS*, volume 31, 2018. 3, 4

[4] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *Proc. ICLR*, 2019. 3, 4

[5] Junnan Li, Caiming Xiong, and Steven CH Hoi. Learning from noisy data with robust representation learning. In *Proc. ICCV*, pages 9485–9494, 2021. 3, 4

[6] Shusen Liu, Dan Maljovec, Bei Wang, Peer-Timo Bremer, and Valerio Pascucci. Visualizing high-dimensional data: Advances in the past decade. *IEEE transactions on visualization and computer graphics*, 23(3):1249–1268, 2016. 6

[7] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. In *Proc. NeurIPS*, volume 33, pages 20331–20342, 2020. 3, 4

[8] Eran Malach and Shai Shalev-Shwartz. Decoupling" when to update" from" how to update". In *Proc. NeurIPS*, volume 30, 2017. 4

[9] Cheng Tan, Jun Xia, Lirong Wu, and Stan Z Li. Co-learning: Learning from noisy labels with self-supervision. In *Proc. ACM MM*, pages 1405–1413, 2021. 3, 4

[10] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *Proc. CVPR*, pages 5552–5560, 2018. 4

[11] Hongxin. Wei, Lei. Feng, Xiangyu Chen, and Bo. An. Combating noisy labels by agreement: A joint training method with co-regularization. In *Proc. CVPR*, 2020. 3, 4

[12] Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. In *Proc. NeurIPS*, volume 33, pages 7597–7610, 2020. 2

[13] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proc. CVPR*, pages 2691–2699, 2015. 4

[14] Kun Yi and Jianxin Wu. Probabilistic end-to-end noise correction for learning with noisy labels. In *Proc. CVPR*, pages 7017–7025, 2019. 4

[15] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption. In *Proc. ICML*, pages 7164–7173, 2019. 4

[16] Sangdoo Yun, Seong Joon Oh, Byeongho Heo, Dongyoon Han, Junsuk Choe, and Sanghyuk Chun. Re-labeling imagenet: from single to multi-labels, from global to localized labels. In *Proc. CVPR*, pages 2340–2350, 2021. 4

[17] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. Mixup: Beyond empirical risk minimization. In *Proc. ICLR*, 2018. 4

[18] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Proc. NeurIPS*, volume 31, 2018. 3, 4

[19] Ganlong Zhao, Guanbin Li, Yipeng Qin, Feng Liu, and Yizhou Yu. Centrality and consistency: two-stage clean samples identification for learning with instance-dependent noisy labels. In *arXiv preprint arXiv:2207.14476*, 2022. 4