RankMatch: Fostering Confidence and Consistency in Learning with Noisy Labels (Supplementary Material)

Ziyi Zhang[1,2], Weikai Chen[3], Chaowei Fang[4], Zhen Li[5], Lechao Chen[6], Liang Lin[2], Guanbin Li[2,7*]
[1]National Key Laboratory of Novel Software Technology, Nanjing University, Nanjing, China
[2]Sun Yat-sen University, Guangzhou, China, [3] Tencent America
[4] Xidian Univiersity, [5] The Chinese University of Hong Kong (Shenzhen), [6] Zhejiang Lab
[7] Research Institute, Sun Yat-sen University, Shenzhen, China
zhangziyi@lamda.nju.edu.cn, liguanbin@mail.sysu.edu.cn

1. Algorithm

Many existing methods [2, 1, 5, 4] alternatively train two networks to combat the confirmation bias. In this paper, we show that simultaneously training two networks still performs well. Specifically, we divide training dataset $\mathcal{D}$ by Eq. 5 and get two clean sets $\mathcal{D}^{(1)}_{\text{cln}}, \mathcal{D}^{(2)}_{\text{cln}}$ by two networks $P(F(x, \theta_i))$ $(i \in \{1, 2\})$ respectively. The final clean set is:

$$\mathcal{D}_{\text{cln}} = \mathcal{D}^{(1)}_{\text{cln}} \cap \mathcal{D}^{(2)}_{\text{cln}}. \tag{1}$$

The remaining samples are regarded as noisy samples: $\mathcal{D}_{\text{nsy}} = \mathcal{D} \setminus \mathcal{D}_{\text{cln}}$. Every image is augmented twice by the two types of augmentation: $(\mathbf{v}^w_i, \mathbf{v}^s_i) = (\mathcal{A}_w(\mathbf{x}_i), \mathcal{A}_s(\mathbf{x}_i))$, $(\mathbf{v'}^w_i, \mathbf{v'}^s_i) = (\mathcal{A}_w(\mathbf{x}_i), \mathcal{A}_s(\mathbf{x}_i))$. Following DivideMix [2], weak augmented images $(\mathbf{v}^w_i, \mathbf{v'}^w_i)$ are leveraged to "co-guess" the correct labels for noisy samples and guide the learning of each network. The full algorithm for implementing RankMatch are shown below.

Algorithm 1: RankMatch. Line 1-5: sample selection by SCV; Line 16: Rank Contrastive Loss

Input: Network $P(\cdot, \theta_1)$ and $P(\cdot, \theta_2)$, training dataset $\mathcal{D}$, augmentation strategies $\mathcal{A}_w$ and $\mathcal{A}_s$, threshold τ, number of prototypes K, number of voters k, ranking parameter r for rank contrastive loss.

1 θ_1, θ_2=WarmUp $(P(\mathcal{D}, \theta_1), P(\mathcal{D}, \theta_2))$
2 **while** $e < MaxEpoch$ **do**
3     $\mathcal{D}^{(1)}_{\text{cln}}$=SCV$(\mathcal{D}, \theta_1, \tau, K, k)$;
4     $\mathcal{D}^{(2)}_{\text{cln}}$=SCV$(\mathcal{D}, \theta_2, \tau, K, k)$;
5   Get $\mathcal{D}_{\text{cln}} = \mathcal{D}^{(1)}_{\text{cln}} \cap \mathcal{D}^{(2)}_{\text{cln}}$ and get $\mathcal{D}_{\text{nsy}}$ by Eq 5;
6   **for** $i = 1$ *to MaxIters* **do**
7   Sample mini-batch $\mathcal{B}$ from $\mathcal{D}_{\text{cln}}$ and $\mathcal{D}_{\text{nsy}}$
8   **for** $(x^{(c)}_b, y^{(c)}_b), (x^{(n)}_b, y^{(n)}_b)$ *in* $\mathcal{B}$ **do**
9   **for** $t = 1, 2$ **do**
10                 $(v^w_b, v^s_b) = (\mathcal{A}_w(x^{(c)}_b), \mathcal{A}_s(x^{(c)}_b))$
11                 $(\tilde{v}^w_b, \tilde{v}^s_b) = (\mathcal{A}_w(x^{(n)}_b), \mathcal{A}_s(x^{(n)}_b))$
12                 $\bar{q}_b = \frac{1}{2}(P(\tilde{v}^w_b, \theta_1) + P(\tilde{v}^w_b, \theta_2))$
13                 $p_b, \hat{y}_b = \max\{\bar{q}_b\}$
14   Obtain $\mathcal{L}_c$ using (v^s_b, y_b) by Eq 6,
15   Obtain $\mathcal{L}_n$ using $(\tilde{v}^s_b, p_b, \hat{y}_b)$ by Eq 8,
16   Obtain $\mathcal{L}_{RCL}$ using $(v^w_b, v^s_b, \tilde{v}^w_b, \tilde{v}^s_b)$ by Eq 11.
17                 $\mathcal{L} = \mathcal{L}_c + \lambda_n \mathcal{L}_n + \mathcal{L}_{RCL} + \mathcal{L}_{div}$
18                 $\theta_t = \text{SGD}(\mathcal{L}, \theta_t)$
19   **end**
20   **end**
21   **end**
22 **end**

Output: θ_1, θ_2.

2. Additional Ablation Study

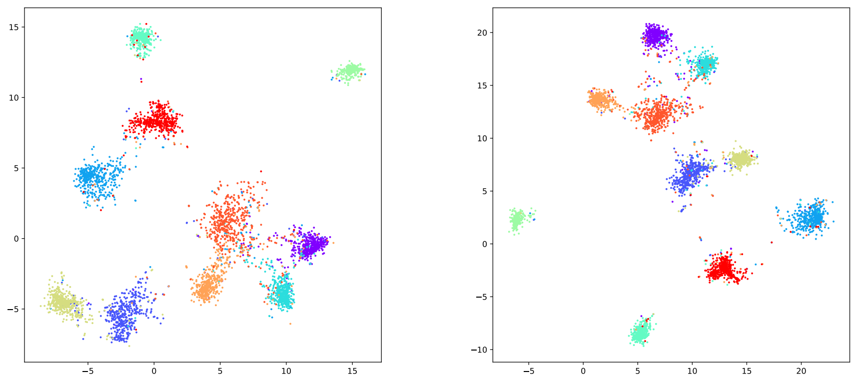
2.1. Rank Contrastive Loss

Our proposed Rank Contrastive Loss (RCL) strengthens the consistency of the similar samples while pushes "dissimilar" samples away, which makes features more discriminative and benefits the sample selection. Thus, we first visualize the features of training images using UMAP [3]. As shown in Fig. 1, features derived by RCL become more compact, and the density of samples around the decision boundaries is reduced, which implies the representations become more discriminative.

2.2. Sensitivity Analysis

SCV introduces three hyperparameters: threshold τ, number of prototypes in each class K and the number of

*Corresponding author.



(a) RankMatch without RCL (b) RankMatch with RCL

Figure 1: Visualization of features learned by RankMatch with or without RCL on CIFAR-100 with 80% label noise. Each variant of RankMatch is trained with 200 epochs. We randomly draw samples of 10 classes from the 100 classes. Compared with representations trained without RCL, full RankMatch generates more discriminative features.

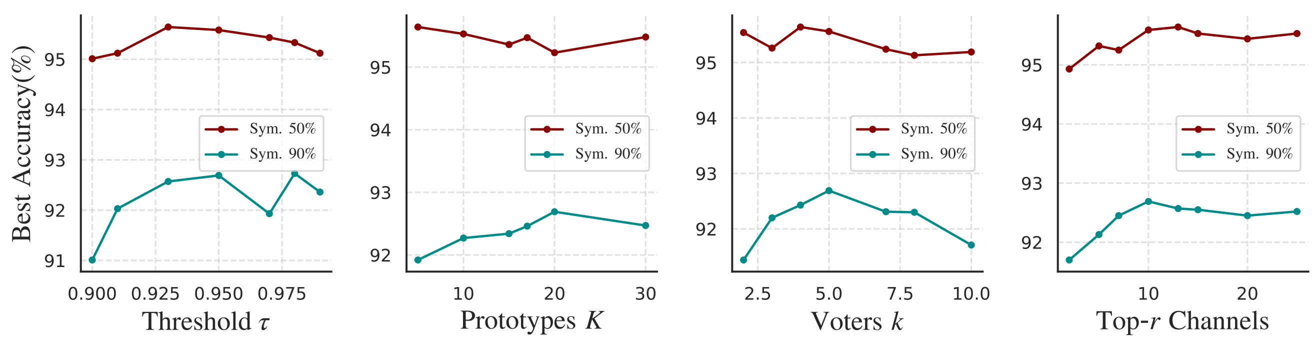


Figure 2: Sensitivity to the variance of hyperparameters. Experiments are conducted on CIFAR-10 under 50% and 90% symmetric noises. We vary the threshold τ from 0.90 to 0.99 to study the effect of confidence in RankMatch. SCV sets up multiple prototypes in each class as confident voting candidates and ranks k-nearest candidates as voters. We range the number of prototypes K from 5 to 30, and range the number of voters k from 2 to 10. The parameter r is used to rank top-r channels for RCL. We range it from 3 to 25.

confident voter k. Rank Contrastive Loss introduces r to select top-r feature channels to set up similarity matrix. Figure 2 illustrates the sensitivity to the variance of these hyperparameters. We can observe that our method is robust against different choices for K, k, r and τ.

Table 1: List of RankMatch hyperparameters for CIFAR

Hyperparameter	CIFAR-10				CIFAR-100			
	20%	50%	80%	90%	20%	50%	80%	90%
λ_n	0.2	1	10	10	0.5	2	5	8

3. Additional Training Details

In the warm-up stage, we only conduct weak augmentation for all experiments. According to observations in the sensitivity analysis, our method is not sensitive to the introduced hyperparameters. Thus, for all benchmarks, we use the same setting of hyperparameters $P = 20$, $k = 5$, $r = 5$, $\tau = 0.95$. For all experiments on CIFAR, we set the training iterations in each epoch as 1024. The only hyperparameter that we tune is the loss weight λ_n for noisy samples. Table 1 shows the value of λ_n that we use.

For both Clothing1M and WebVision, we train the network using SGD as in DivideMix [2], and a batch size of 32. The warm up period occupies four epochs, and the loss weight λ_n is set as 0. For Clothing1M, we set the number of iterations in each epoch as 1000, and train networks for

80 epochs. We initialize the learning rate as 0.003 and reduce it by a factor of 10 after 40 epochs. For WebVision, we train the network for 100 epochs, and reduce the initialized learning rate 0.01 by a factor of 10 after 50 epochs.

4. Additional Experimental Results

Table 2: Average test accuracy (%) on CIFAR-10 dataset over the last 10 epochs. We run our method three times with different random seeds and report the mean and the standard deviation.

Noise type	Sym.				Asym.
Noise level	20%	50%	80%	90%	40%
Cross-Entropy	82.7	57.9	26.1	16.8	85.0
DivideMix	95.0	93.7	92.4	74.2	91.4
RankMatch	**96.43**$\pm$0.08	**95.39**$\pm$0.09	**94.26**$\pm$0.11	**92.01**$\pm$0.07	**94.36**$\pm$0.25

We run our method for three times with different random seeds and report the mean and the standard deviation in Table 2. Compared with the DivideMix [2], our method outperforms most recent methods by a large margin and with small standard deviation, which implies that our experimental results are statistically significant.

References

[1] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, pages 8536–8546, 2018.

[2] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *ICLR*, 2020.

[3] L. McInnes, J. Healy, and J. Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv e-prints*, Feb. 2018.

[4] Kento Nishi, Yi Ding, Alex Rich, and Tobias Höllerer. Augmentation Strategies for Learning with Noisy Labels. *arXiv e-prints*, page arXiv:2103.02130, Mar. 2021.

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