Robust Collaborative Learning with Noisy Labels

Mengying Sun*, Jing Xing†, Bin Chen†, Jiayu Zhou*
*Computer Science and Engineering, Michigan State University, East Lansing, MI, USA
†Pediatrics and Human Development, Pharmacology and Toxicology, Michigan State University, Grand Rapids, MI, USA
Emails: sunmeng2@msu.edu, xingjin1@msu.edu, chenbi12@msu.edu, jiayuz@msu.edu

***Abstract*—Learning with curriculum has shown great effectiveness in tasks where the data contains noisy (corrupted) labels, since the curriculum can be used to re-weight or filter out noisy samples via proper design. However, obtaining curriculum from a learner itself without additional supervision or feedback deteriorates the effectiveness due to sample selection bias. Therefore, methods that involve two or more networks have been recently proposed to mitigate such bias. Nevertheless, these studies utilize the collaboration between networks in a way that either emphasizes the disagreement or focuses on the agreement while ignores the other. In this paper, we study the underlying mechanism of how disagreement and agreement between networks can help reduce the noise in gradients and develop a novel framework called Robust Collaborative Learning (RCL) that leverages both disagreement and agreement among networks. We demonstrate the effectiveness of RCL on both synthetic benchmark image data and real-world large-scale bioinformatics data.**

***Index Terms*—curriculum learning; robust learning; weak supervision**

I. Introduction

Recent years have witnessed huge successes of supervised learning using deep neural networks in various domains [1], [2]. One decisive factor behind such successes is the availability of a sufficiently large amount of training data [3]. In cases where obtaining accurate labels is too expensive, practitioners could use affordable apparatuses to collect less reliable *noisy* labels, e.g., the online crowd-sourcing tool Mechanical Turk. Besides, labels from high-throughput experiments (e.g., biological profiling and chemical screening) often contain inevitable noise due to technical and biological variations.

Learning with noisy labels has imposed additional challenges. Sometimes the data quality is known *a priori* [4]–[6], but a more common scenario is that, the data available is a mixture of samples with both clean and noisy labels and one does not know, or only has partial knowledge of the underlying distribution of the noise [7]–[10]. In this problem setting, a learning process that is aware of noise in the labels and actively mitigates the negative impacts from the noisy labels, is the key to improving the generalization of learned models.

Learning with curriculums, eg., self-paced learning (SPL) [11]–[13] reveals its power in dealing with noisy data. The reason is that noisy samples can be re-weighted or even filtered out via proper curriculum designs. It has been proved by [14] that the latent objective of self-paced learning is equivalent to a robust loss function, which also shed lights on the effectiveness of SPL on noisy data. However, a major drawback of determining curriculum based on the learner's own ability is the sample selection bias from the learner itself. The error made in early stage will be enhanced as training proceeds. Therefore, two or more networks have been introduced in recent works to mitigate the selection bias [15], [16]. Nevertheless, these studies either emphasize the disagreement or focus on the agreement between networks without considering the other. Therefore in this paper, we propose a novel framework called **R**obust **C**ollaborative **L**earning (**RCL**) to deal with noisy labels. The main contributions of this paper are:

- We show *disagreement* between networks can diversify the gradients of model weights from noisy samples, which slows down the accumulation of noisy gradients.
- We show that under certain conditions, *agreement* from more than one network can improve the quality of data selection, i.e., the label purity increases.
- Combining the above two findings, we propose RCL framework that consists of multiple networks, where each network is an individual learner and exchanges knowledge with its *Peer* system. The knowledge of the *Peer* system is fused from multiple networks, by adaptively encouraging *disagreement* in the early stage and *agreement* in the later stage, which fully boost the selection of clean samples for training.

We demonstrate RCL on both synthetic and real experiments. For synthetic experiment, we use the benchmark data CIFAR10 and CIFAR100 [17] under different noise settings following literature [15], [18]. We further validated our framework on cancer drug development using large-scale genomic data [19], [20]. The proposed method achieves state-of-art performance and significantly outperforms baselines in large noise settings, on both image and bioinformatics data.

II. Related Work

Our work originates from curriculum learning and its following variants, and also connects to weak supervision and robust learning. Below are the related works for each direction.

Inspired by the fact that humans learn better when trained with a curriculum-like strategy, [21] first proposed curriculum learning. Results on both visual and language tasks have shown that training on easy tasks first and then hard tasks led to faster convergence as well as better generalization. Instead of using a specified curriculum, [11] incorporated a latent variable associated with each sample, and jointly optimized the model parameters and the curriculum. Later, a variety of approaches with different predefined curriculums were proposed and validated [12], [13], [22]–[24]. Besides, instead of using predefined curriculum, [25] proposed to learn

2374-8486/20/$31.00 ©2020 IEEE
DOI 10.1109/ICDM50108.2020.00163

a data-driven curriculum when auxiliary data is available. The authors also designed an efficient algorithm for training very deep neural networks with curriculum. Following that, [15] proposed co-teaching framework, a system of two networks that exchange selected samples to alleviate bias brought by one network. Co-teaching [15] works well empirically with several follow-up works [26], [27]. [28] later proposed to make use of the unselected samples by correcting their labels and combining them with selected samples for training. Other very recent works also aggregated knowledge from multiple sources, e.g., multiple networks or multiple training epochs of a single network to filter out noisy data [16], [29].

Learning with corrupted labels also relates to weak supervised learning, and its recent advances can be summarized into the following groups. A common way to leverage weak labels when the quality of data is known *a priori* is to use the pre-train and fine-tune scheme based on the amount of clean and weak data [4], [5], [30]. Another line of methods design surrogate loss functions for robust learning [7], [8], [31]. Some approaches model the noise pattern or estimate the error transition matrix [9], [32], and denoise by either adding an extra layer [33], or using generative models [10], [34]. Other methods utilize semi-supervised learning techniques [35], [36] to revise weak labels for further training [37], or regularize the learning procedure [38]. Recently, learning-to-learn methods have also been proposed to tackle such problems by manipulating gradient update rules [6], [39]. Since our work mainly follows curriculum learning, we do not expose further details here and refer readers of interest to the original papers.

III. Background and Investigation

A. A Revisit of Self-paced Learning (SPL)

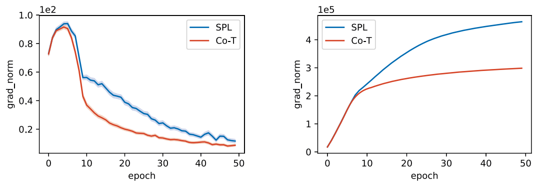
SPL [11] introduces a latent variable associated with each training sample, and solves them during training. Denoting the latent variables in a vector $\mathbf{v} \in [0,1]^n$, where n is the sample size, the objective function of SPL can be written as:

$$\min_{\mathbf{w},\mathbf{v}\in[0,1]^n} \mathbf{E}(\mathbf{w},\mathbf{v},\lambda) = \sum\nolimits_{i=1}^{n} v_i L\big(y_i, f(\mathbf{x}_i,\mathbf{w})\big) + g(v_i,\lambda)$$

where $g(v,\lambda)$ is the curriculum function and regularizes the weight of a given sample, λ is a control parameter. $\mathbf{w}$ and $\mathbf{v}$ are optimized alternatively [40]. A simple example of g can be $g(v,\lambda) = -\lambda v$ with closed-form solution for v at each step: $v^*(\lambda; l) = 1$ when $l < \lambda$ and $v^*(\lambda; l) = 0$ otherwise, where l is the loss for one sample. The design of $g(v,\lambda)$ reveals the nature of SPL: if a sample has large loss on the current model, it is likely to be more difficult to learn or even an outlier.

B. The power of Disagreement

A major drawback of SPL is the sample selection bias from one learner. The error that takes place in the early stage will be reinforced as training continues. To mitigate this, a second network is introduced in co-teaching [15], where two networks exchange selected samples to train. Such strategy works better than SPL in practice with the underlying mechanism not well studied. Here, we show that exchanging data introduces disagreement between two networks, which can diversify noisy gradients and lead to higher gradient purity.



(a) GN at each epoch (b) Accumulative GN

Fig. 1: Gradient norm (GN) of *noisy* data on CIFAR10. Co-teaching learning process has less impact from noisy data.

In SGD, the gradient update for each mini-batch is:

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta^t \frac{1}{n_r} \sum\nolimits_{j=1}^{n_r} \nabla l_j(\mathbf{w}^t) = \mathbf{w}^t - \eta^t \frac{1}{n_r} S_\nabla,$$

where n_r is the batch size, η^t the step size. Considering each of the two networks, with an oracle that provides the ground truth whether a label is noisy or not, we can decompose gradient summation S_∇ into four disjoint components based on data quality (clean or noisy) and network agreement (agree or not):

$$\begin{aligned} S_\nabla = & \underbrace{\Sigma_{j\in I_{11}} \nabla l_j(\mathbf{w}^t)}_{\text{agreed clean}} + \underbrace{\Sigma_{j\in I_{10}} \nabla l_j(\mathbf{w}^t)}_{\text{disagreed clean}} \\ & + \underbrace{\Sigma_{j\in I_{01}} \nabla l_j(\mathbf{w}^t)}_{\text{agreed noisy}} + \underbrace{\Sigma_{j\in I_{00}} \nabla l_j(\mathbf{w}^t)}_{\text{disagreed noisy}}, \end{aligned} \quad (1)$$

In co-teaching, the set I_{00} (noisy, disagree) are diversified by exchanging data points between the two networks. Such disagreement is crucial and can cause several effects. First, the gradient norm of noisy data may diminish; second, the diversification can take effect across time since gradients are eventually summed and applied to network parameters; third, introducing disagreement is equivalent to adding small perturbations on network parameters, which could increase the robustness of the network.

We evaluate these effects in a small synthetic experiment. Given an image-classification problem, e.g., CIFAR10, for each class, we manually flip 45% labels into the adjacent class. Then we compare the gradients of noisy samples between SPL and co-teaching, i.e., $\Sigma_{j\in I_{01}} \nabla l_j(\mathbf{w}^t) + \Sigma_{j\in I_{00}} \nabla l_j(\mathbf{w}^t)$ in Eq. (1). The gradients are calculated from the last linear layer of a CNN model. We can see that disagreement from exchanging data helps achieve smaller noisy gradient compared to SPL and it also slows down the accumulation of noisy gradients.

C. The power of Agreement

When the learners are mature, aggregating their knowledge can be beneficial as compared to only exchanging them. In fact, recent works [16], [29] propose to aggregate knowledge from multiple networks to filter out noisy samples during training and show promising results. However, ensemble does not guarantee higher purity especially in the early training stage since errors could also be magnified. Therefore, a common strategy is to train the entire data until certain epochs and then perform the ensemble. Nevertheless, such strategy may not be optimal especially when the noise rate is large.

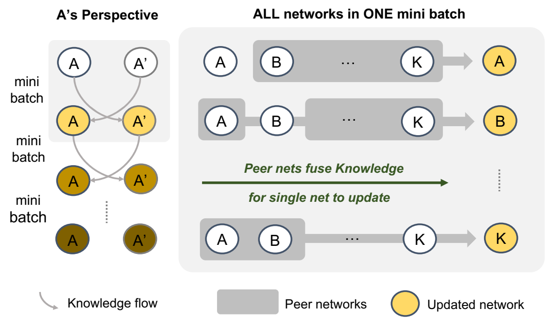


Fig. 2: Robust Collaborative Learning (RCL) framework. Left: one network's view, A' denotes all the peer networks of A; Right: all networks in one mini batch. Knowledge fusion and update can be done in parallel for all the networks.

IV. Method

Fig. 2 shows the overall structure of the proposed method. In RCL, each network is an individual learner, while the rest networks form a *Peer* system. From each network's perspective, it exchanges knowledge with its *Peer* (Fig. 2 Left). The knowledge it receives is fused from multiple networks in the *Peer* system, while the knowledge it offers will wait for fusion when itself is served as a peer network (Fig. 2 Right). The pseudo code of the overall algorithm is illustrated in Alg. 1. Next we introduce each component of RCL in detail.

A. Self-Knowledge

During one mini-batch, each network first selects top $R \times 100\%$ ranked small-loss samples. Due to the memorization effect, i.e., deep neural networks tend to learn easy patterns first and then memorize noise at later epochs [41], the reserve rate $R(T)$ is designed to be monotonically decreasing w.r.t. epoch T from 100% until it reaches clean rate $(1-\epsilon)\times 100\%$, where ϵ is the noise rate. T_{cut} is the switch epoch, after which, only $(1-\epsilon)\times 100\%$ of the data will be selected. The selected samples are the self-knowledge of each individual network, and will be used in knowledge fusion step.

B. Knowledge Fusion

For a given network k, it utilizes knowledge from its *Peer* system, which includes all the rest networks except for network k. The knowledge of this system is fused from multiple networks via a knowledge fusion function. Ideally, when the networks are trained well, the knowledge of agreement, i.e., data points picked by all the peer networks can be used to update network k. However, during the early stage, the networks are prone to making mistakes. Therefore, disagreement is introduced to reduce the noise in gradients. There are two parameters associated with it. α determines the switch epoch between disagreement and agreement. Instead of setting a particular epoch, we design α as the lag of switch epoch compared to T_{cut}. Fusion rate r controls the strength of disagreement, i.e., the proportion of disagreed samples that will be included in addition to the common samples. As epoch increases, less disagreed samples will be included

Algorithm 1: Pseudo code for RCL Algorithm.

```
Input: $K$ networks $\{\Theta_1..\Theta_K\}$, training data $\mathcal{D}$, noise rate $\epsilon$;
(Fixed) learning rate $\eta$, epoch $T_{\max}$ and iteration $N_{\max}$;
(Hyper) epoch $T_{\text{cut}}$, fusion multiplier $\alpha$, fusion exponent $\beta$.
Output: Updated network parameters $\{\Theta'_1..\Theta'_k\}$
1  for $T = 1$ to $T_{\max}$ do
2      Shuffle training set $\mathcal{D}$
3      Update $R(T) = 1 - \epsilon \cdot \min\left\{\frac{T}{T_{\text{cut}}}, 1\right\}$    // remember rate
4      Update $r(T) = 1 - \min\left\{\left(\frac{T}{\alpha T_{\text{cut}}}\right)^{\beta}, 1\right\}$    // fusion rate
5      for $N = 1$ to $N_{\max}$ do
6          Fetch mini-batch $D$ from $\mathcal{D}$
7          for $k = 1$ to $K$ do
8              // pick top $R(T)$ small-loss instances
9              Obtain $D_k = \arg\min_{\mathbb{D}:|\mathbb{D}|\le R(T)|D|} l(f_{\Theta_k}, \mathbb{D})$
10         for $k = 1$ to $K$ do
11             // integrate knowledge from all peer networks
12             Obtain $D'_k = Knowledge\left(D_{\{1..K\}\setminus k}, r(T)\right)$
13             // update network $k$
14             Update $\Theta'_k = \Theta_k - \eta\nabla l(f_{\Theta_k}, D'_k)$
15 return $\Theta' = \{\Theta'_1..\Theta'_k\}$;
```

Algorithm 2: Knowledge Fusion Function.

```
Input: Given the $k$-th network, the knowledge of all other
       peer networks $\{D_1..D_K\} \setminus D_k$; Fusion rate $r(T)$.
Output: Data for updating the $k$-th network $D'_k$.
1  $D_{\text{agree}} = \text{Intersect}\left(\{D_1..D_K\} \setminus D_k\right)$
2  $D_{\text{potential}} = \text{Union}\left(\{D_1..D_K\} \setminus D_k\right)$
3  if $|D_{agree}| == |D_{potential}|$ then
4      $D'_k = D_{\text{agree}}$
5  else
6      $D_{\text{uncertain}} = D_{\text{potential}} - D_{\text{agree}}$
7      $n_{\text{in}} = r(T) \cdot |D_{\text{uncertain}}|$
8      $D_{\text{in}} = \text{random\_sample}\left(D_{\text{uncertain}}, n_{\text{in}}\right)$
9      $D'_k = D_{\text{agree}} + D_{\text{in}}$
10 return $D'_k$;
```

until only common ones are selected. The decay of strength of disagreement is controlled by a hyper-parameter β. In summary, the fusion rate at each epoch is calculated as:

$$r(T) = \begin{cases} 1 - \left(T/(\alpha T_{\text{cut}})\right)^{\beta}, & \text{if } T < \alpha T_{\text{cut}} \\ 0, & \text{if } T \ge \alpha T_{\text{cut}} \end{cases} \quad (2)$$

After that, $r \times 100\%$ of the disagreed samples are randomly picked and added to the agreed samples as the final knowledge of the *Peer* system. Such randomness also introduces certain level of disagreement since each network will receive different candidates to train even if the common samples are the same within each *Peer* system. The pseudo code of knowledge fusion procedure for multiple networks is illustrated in Alg. 2.

C. Knowledge Update

Network k receives the candidate samples from its *Peer* system and update parameters based on them. The same procedure can be done in parallel for all the networks. After all the networks have been updated, they enter the next iteration.

Method	Standard	SPL	De-CP	Co-T	K=3	K=5	K=7	K=9	K=11	K=13	+R	p-val	# nets
Data Noise							Test Accuracy						
CIFAR10 SYM 50%	48.52	70.92	45.53	73.45	75.72	77.44	78.01	78.47	**78.91**	78.91	**7.43**	2e-9	11
CIFAR10 PF 45%	48.65	56.08	49.24	72.77	74.59	76.28	77.28	78.25	78.70	**79.15**	**8.77**	<1e-9	13
CIFAR100 SYM 50%	21	36.21	17.51	38.14	40.05	41.83	42.43	42.96	**43.77**	43.14	**14.75**	4e-07	11
CIFAR100 PF 45%	29.57	28.63	26.17	30.44	32.51	34.90	36.86	37.85	39.02	**39.15**	**28.58**	4e-09	13
Data Noise							Pure Ratio						
CIFAR10 SYM 50%	50.31	84.22	40.48	83.95	86.96	88.82	89.47	89.86	90.11	**90.33**	**7.33**	<1e-9	11
CIFAR10 PF 45%	54.89	68.09	51.23	79.25	82.72	84.99	86.16	87.17	87.69	**88.11**	**11.18**	<1e-9	13
CIFAR100 SYM 50%	49.95	80.51	42.89	80.65	83.85	86.20	87.14	87.82	**88.20**	88.20	**9.36**	<1e-9	11
CIFAR100 PF 45%	54.84	58.66	53.42	59.03	61.07	63.94	66.97	68.65	69.36	**69.99**	**18.57**	<1e-9	13

TABLE I: Performance of non-ensemble baselines and the proposed method RCL over different number of networks (K) on fixed noise rates. Average over 5 random seeds. +R denotes relative improvement. Significance t-tests (one-side) are conducted between RCL and the best baseline. A p-value less than 0.05 is considered as significant difference.

V. Experiment

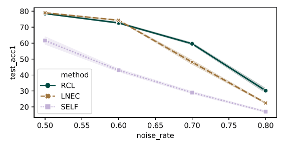
A. Synthetic Experiment

Datasets. We use CIFAR-10 and CIFAR-100 datasets and manually create corrupted labels following the strategy in [15], [18]. Two common noise scenarios are considered, symmetric flip (SYM) and pair flip (PF). For SYM, the label of each class is uniformly random flipped to the rest classes with equal probability; for PF, the label of each class only flips to one different but similar class. The noise rate ϵ quantifies the overall proportion of labels that are flipped for each class.

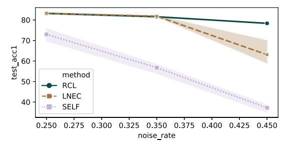
Network Architecture. We follow the 9-layer CNN architecture from [15], [42]. We use Adam optimizer with a momentum of 0.9 and an initial learning rate of 0.001. The batch size is 128 and the maximum epoch is 200. We implemented and run the models using PyTorch and NIVIDIA GPUs.

Experimental Setup We keep the major hyper-parameter of co-teaching [15], T_{cut} in Alg. 1, and fix those having subtle effect on results in the original paper. In our experiments, we first tune $T_{\text{cut}} = \{5, 10, 15\}$ for one and two networks (i.e., SPL and Co-teaching). Then we use the best for further tuning RCL. For other hyper-parameters, we fix $\alpha = 2$ for all scenarios and use fixed β for a given noise scenario, based on the sensitivity analysis using $\beta = \{0.0, 0.1, 0.3, 0.5, 1.0, 2.0, 8.0\}$. We test over a range of number of networks $K = \{3, 5, 7, 9, 11, 13\}$, and noise rates $\epsilon = \{0.25, 0.35, 0.45\}$ for pairflip noise and $\epsilon = \{0.5, 0.6, 0.7, 0.8\}$ for symmetric noise.

Baselines. (1) Standard, a single network trained on the entire dataset. (2) SPL [25], a single network that produces curriculum based on its own. (3) Decoupling (De-CP) [43], a double-net system where the networks only updates parameters from data whose prediction label is disagreed between two networks. (4) Co-teaching [15], a double-net system in which the two networks exchange curriculum at each iteration. (5) Ensemble consensus [16], specifically the LNEC variant, a multi-net system that explores agreement between multiple networks. (6) Self-ensemble (SELF) [29], which explores agreement between consecutive epochs within a network. The original implementation involves other hybrid components without code release, we adopt the core idea of temporal ensemble and implement the method. The test accuracy is evaluated on *clean* test set. The pure ratio measures the average proportion of clean data that is selected by the algorithm during training. All metrics are evaluated on *one* network.



(a) CIFAR10 symmetric flip



(b) CIFAR10 pair flip

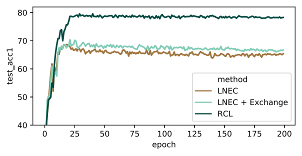
Fig. 3: Test accuracy of ensemble methods and RCL (K=9) over various noise rates on CIFAR10 (avg. 3 runs).

Data	Noise	Test accuracy			Pure ratio		
		SELF	LNEC	RCL	SELF	LNEC	RCL
CF10 SYM	50%	61.65	79.00	78.51	81.38	90.07	89.85
	60%	42.97	**74.32**	72.65	64.89	**86.15**	85.14
	70%	28.99	48.05	**59.60**	46.20	61.31	**74.40**
	80%	17.04	22.45	**30.20**	29.13	31.65	**42.04**
CF10 PF	25%	72.97	83.23	83.18	88.34	93.13	92.75
	35%	56.77	81.83	81.57	75.58	91.05	90.93
	45%	37.21	63.09	**78.34**	57.61	72.22	**87.28**
CF100 SYM	50%	25.02	43.89	43.37	66.69	86.92	**87.99**
	60%	15.82	32.07	**36.12**	52.60	78.06	**82.20**
	70%	7.69	23.41	**26.71**	37.23	64.44	**72.34**
	80%	3.29	11.99	**14.79**	24.02	40.20	**49.96**
CF100 PF	25%	38.23	**52.83**	51.03	82.21	**92.04**	87.94
	35%	28.08	**46.49**	45.91	69.12	**82.28**	81.28
	45%	19.28	31.48	**38.38**	53.99	61.16	**69.17**

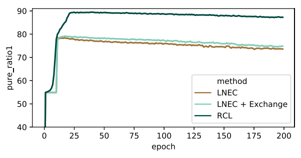
TABLE II: Final performance of ensemble methods and RCL over various noise rates (K=9). Bold numbers indicate that the method is significantly better than the second best method.

Benefit of agreement. The benefit of agreement comes from ensemble of multiple networks on the selection of clean samples, which can be verified by adding number of networks while fixing other components. Table I shows the results of RCL over different number of networks for a given noise rate. We can see that test accuracy improves as the number of networks increases. Higher pure ratio generally leads to higher test accuracy. Importantly, RCL selects significantly more clean data compared to baselines.

Benefit of disagreement. The agreement may not work when the noise rate is large, especially during the early training stage, since it also ensembles the error. In such situation, disagreement can help reduces the noise in gradients and helps pick out more clean samples. The disagreement takes place in terms of two levels: the first level is to exchange data in a learn-from-the-other way such that each network receives different data from the its *Peer* system, the second level is that the strength of disagreement varies along the

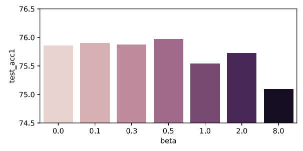


(a) Test accuracy

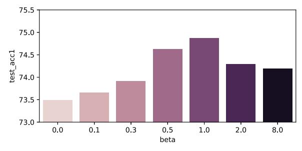


(b) Pure ratio

Fig. 4: Improvement of RCL over ensemble method by each *disagreement* level on CIFAR10 PF 45% scenario (K=9).



(a) symmetric 50%



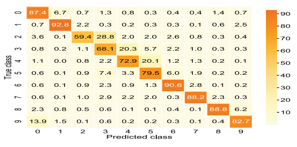
(b) pairflip 45%

Fig. 5: Test accuracy of RCL on various βs (avg. 3 runs, K=3).

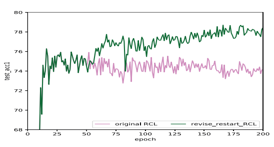
training procedure and can be controlled by a hyper-parameter. Both levels can improve the selection of clean samples as well as the generalization performance. To verify these, we compare RCL with several ensemble methods [16], [29] which only explore agreement among multiple networks (or multiple training epochs), when all the networks use the same set of candidates to train. Fig. 3 shows the test accuracy of the competing methods for different noise rates on CIFAR10. Pure ratio reveals the exact same pattern and therefore is not shown in the figure. Complete results are presented in Table II. First, temporal ensemble (SELF) does not work as good as network ensembles in the provided scenarios. Second, when the noise rate is small, RCL reaches similar accuracy as the state-of-art ensemble methods; when the noise rate is large, RCL yields significantly better performance compared to the ensemble baselines. We also find that the variations of baselines are much larger compared to RCL, which indicates the robustness of RCL. Next, in order to see the improvement brought by each level of disagreement, we add the data exchange step onto LNEC baseline and compare it with pure LNEC and RCL (all use 9 networks). The result is shown in Fig. 4.

Sensitivity analysis The parameter β controls the strength of disagreement and is important in RCL. We test over a range of different values to study its behavior. Fig. 5 shows the results on CIFAR10. We can see the test accuracy shows opposite patterns for different noise scenarios. When the task is relatively easy (e.g., CIFAR10 symmetric 50%), small β yields better accuracy. If the noise rate is large, it favors relatively large β which encourages disagreement during the early stage. However, extreme large value of β is not beneficial.

Reduce time complexity While being effective, RCL requires more computational power and running time compared to methods using one or two networks. One way to reduce time complexity without deteriorating the performance is to utilize the unselected samples during training. Therefore, we propose a revise-and-restart strategy based on the current framework. When the training reaches certain epochs (usually plateau), we first revise the labels of the unselected samples based on the current prediction. Then we restart the training procedure, i.e., first introduce disagreement and then agreement. Fig. 6



(a) Confusion matrix.



(b) Test accuracy

Fig. 6: Revise and restart RCL on CIFAR10 PF 45% (K=3). Left: confusion matrix after revision. Numbers in cells denote percentages. Right: test accuracy w.r.t epochs.

shows the result on CIFAR10 by using only 3 networks. After revising the labels of unselected samples at epoch 50, the label precision for each class is shown in Fig. 6a, which is significantly higher compared to 55%. Fig. 6b shows the episode curve of revise and restart compared to the original 3 networks. The performance of revise-and-restart RCL using 3 networks reaches as high as that of 9 networks, which is a considerable reduction on the computational burden.

B. *Drug-induced Gene-Expression Change Prediction*

We apply RCL to a real-world problem in the bioinformatics domain, where the noise naturally exists and the noise rate is **unknown** (tuned as a hyper-parameter).

Due to space limit, we briefly introduce the experimental settings. The task is to predict the responses of 11,000 drugs on 978 genes (down/up-regulate and no change, 3 classes) for 6 cell lines, whose results can be used for cancer drug discovery [19], [44]. The original responses are continuous values and each drug profile is repeatedly tested for different number of times. Some drugs profiles have consistent readings while others do not, indicating different label qualities. We obtain a small subset of high-quality profiles following standard statistical procedure [19] to serve as test set and use the remaining data for training. The features and network architecture follow [20]. Table III shows the results in which RCL significantly outperforms baselines in most cases.

VI. Conclusion

In this paper, we proposed a novel deep framework RCL for learning with noisy labels. Both synthetic and real experiments demonstrate the power of RCL. While being effective, RCL deserves further exploration and perfection in the future.

Acknowledgment

This research is supported in part by NSF IIS-1749940 (JZ), ONR N00014-17-1-2265 (JZ), NIH 1R01GM134307 (JZ, BC), NIH K01ES028047 (BC).

References

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
[3] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 843–852.
[4] X. Liang, S. Liu, Y. Wei, L. Liu, L. Lin, and S. Yan, "Towards computational baby learning: A weakly-supervised approach for object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 999–1007.

Cell	Standard		SPL		Decoupling		Co-teaching		SELF		LNEC		RCL		p-val	Best	
Line	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	K	FR
VCAP	47.64	0.462	49.36	0.473	47.70	0.473	50.26	0.482	49.23	0.474	49.12	0.472	**52.22**	**0.495**	**1e-3**	4	0.3
MCF7	51.41	0.446	55.14	0.466	54.57	0.456	56.44	0.476	56.45	0.475	54.04	0.464	**57.98**	**0.482**	**5e-4**	4	0.2
PC3	47.90	0.474	47.92	0.473	45.32	0.467	48.22	0.477	48.42	0.478	48.18	0.475	**49.04**	**0.484**	**8e-3**	4	0.1
A549	50.73	0.403	53.52	0.417	53.38	**0.422**	52.15	0.414	53.67	0.421	53.63	0.417	**54.00**	0.421	0.25	3	0.1
A375	46.42	0.396	47.14	0.395	**49.37**	**0.407**	48.87	0.399	48.43	0.402	46.95	0.393	48.99	0.395	-	3	0.4
HT29	46.39	0.441	46.51	0.444	47.18	0.449	47.86	0.456	47.12	0.453	46.66	0.452	**48.13**	**0.458**	**0.05**	3	0.3

TABLE III: Final performance for six cell lines (avg. 5 runs). K = number of networks, FR = forget rate. Significance t-tests (one-side) are conducted between RCL and the best baseline. A p-value less than 0.05 is considered as significant difference.

[5] A. Severyn and A. Moschitti, "Unitn: Training deep convolutional neural network for twitter sentiment classification," in *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, 2015, pp. 464–469.

[6] M. Dehghani, A. Severyn, S. Rothe, and J. Kamps, "Learning to learn from weak supervision by full supervision," *arXiv preprint arXiv:1711.11383*, 2017.

[7] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari, "Learning with noisy labels," in *Advances in neural information processing systems*, 2013, pp. 1196–1204.

[8] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[9] A. Menon, B. Van Rooyen, C. S. Ong, and B. Williamson, "Learning from corrupted binary labels via class-probability estimation," in *International Conference on Machine Learning*, 2015, pp. 125–134.

[10] P. Varma, B. He, D. Iter, P. Xu, R. Yu, C. De Sa, and C. Ré, "Socratic learning: Correcting misspecified generative models using discriminative models," *arXiv preprint arXiv:1610.08123*, 2017.

[11] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Advances in Neural Information Processing Systems*, 2010, pp. 1189–1197.

[12] J. S. Supancic and D. Ramanan, "Self-paced learning for long-term tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2379–2386.

[13] L. Jiang, D. Meng, S.-I. Yu, Z. Lan, S. Shan, and A. Hauptmann, "Self-paced learning with diversity," in *Advances in Neural Information Processing Systems*, 2014, pp. 2078–2086.

[14] D. Meng, Q. Zhao, and L. Jiang, "What objective does self-paced learning indeed optimize?" *arXiv preprint arXiv:1511.06049*, 2015.

[15] 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*, 2018, pp. 8527–8537.

[16] J. Lee and S.-Y. Chung, "Robust training with ensemble consensus," *arXiv preprint arXiv:1910.09792*, 2019.

[17] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[18] B. Van Rooyen, A. Menon, and R. C. Williamson, "Learning with symmetric label noise: The importance of being unhinged," in *Advances in Neural Information Processing Systems*, 2015, pp. 10–18.

[19] A. Subramanian, R. Narayan, S. M. Corsello, D. D. Peck, T. E. Natoli, X. Lu, J. Gould, J. F. Davis, A. A. Tubelli, J. K. Asiedu *et al.*, "A next generation connectivity map: L1000 platform and the first 1,000,000 profiles," *Cell*, vol. 171, no. 6, pp. 1437–1452, 2017.

[20] G. Woo, M. Fernandez, M. Hsing, N. A. Lack, A. D. Cavga, and A. Cherkasov, "Deepcop: deep learning-based approach to predict gene regulating effects of small molecules," *Bioinformatics*, 2019.

[21] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 41–48.

[22] Y. J. Lee and K. Grauman, "Learning the easy things first: Self-paced visual category discovery," in *CVPR 2011*. IEEE, 2011, pp. 1721–1728.

[23] L. Jiang, D. Meng, Q. Zhao, S. Shan, and A. G. Hauptmann, "Self-paced curriculum learning," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[24] D. Zhang, D. Meng, and J. Han, "Co-saliency detection via a self-paced multiple-instance learning framework," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 5, pp. 865–878, 2016.

[25] 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.

[26] X. Yu, B. Han, J. Yao, G. Niu, I. W. Tsang, and M. Sugiyama, "How does disagreement benefit co-teaching?" *arXiv preprint arXiv:1901.04215*, 2019.

[27] X. Wang, S. Wang, J. Wang, H. Shi, and T. Mei, "Co-mining: Deep face recognition with noisy labels," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9358–9367.

[28] H. Song, M. Kim, and J.-G. Lee, "Selfie: Refurbishing unclean samples for robust deep learning," in *International Conference on Machine Learning*, 2019, pp. 5907–5915.

[29] D. T. Nguyen, C. K. Mummadi, T. P. N. Ngo, T. H. P. Nguyen, L. Beggel, and T. Brox, "Self: Learning to filter noisy labels with self-ensembling," *arXiv preprint arXiv:1910.01842*, 2019.

[30] J. Deriu, A. Lucchi, V. De Luca, A. Severyn, S. Müller, M. Cieliebak, T. Hofmann, and M. Jaggi, "Leveraging large amounts of weakly supervised data for multi-language sentiment classification," in *Proceedings of the 26th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 2017, pp. 1045–1052.

[31] H. Masnadi-Shirazi and N. Vasconcelos, "On the design of loss functions for classification: theory, robustness to outliers, and savageboost," in *Advances in neural information processing systems*, 2009, pp. 1049–1056.

[32] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus, "Training convolutional networks with noisy labels," *arXiv preprint arXiv:1406.2080*, 2014.

[33] J. Goldberger and E. Ben-Reuven, "Training deep neural-networks using a noise adaptation layer," 2016.

[34] A. J. Ratner, C. M. De Sa, S. Wu, D. Selsam, and C. Ré, "Data programming: Creating large training sets, quickly," in *Advances in neural information processing systems*, 2016, pp. 3567–3575.

[35] S. Abney, "Understanding the yarowsky algorithm," *Computational Linguistics*, vol. 30, no. 3, pp. 365–395, 2004.

[36] M. Culp and G. Michailidis, "An iterative algorithm for extending learners to a semi-supervised setting," *Journal of Computational and Graphical Statistics*, vol. 17, no. 3, pp. 545–571, 2008.

[37] J. Han, P. Luo, and X. Wang, "Deep self-learning from noisy labels," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5138–5147.

[38] A. Vahdat, "Toward robustness against label noise in training deep discriminative neural networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 5596–5605.

[39] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas, "Learning to learn by gradient descent by gradient descent," in *Advances in neural information processing systems*, 2016, pp. 3981–3989.

[40] F. Vaida, "Parameter convergence for em and mm algorithms," *Statistica Sinica*, vol. 15, no. 3, p. 831, 2005.

[41] D. Arpit, S. Jastrzębski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio *et al.*, "A closer look at memorization in deep networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 233–242.

[42] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," *arXiv preprint arXiv:1610.02242*, 2016.

[43] E. Malach and S. Shalev-Shwartz, "Decoupling" when to update" from" how to update"," in *Advances in Neural Information Processing Systems*, 2017, pp. 960–970.

[44] B. Chen, L. Ma, H. Paik, M. Sirota, W. Wei, M.-S. Chua, S. So, and A. J. Butte, "Reversal of cancer gene expression correlates with drug efficacy and reveals therapeutic targets," *Nature communications*, vol. 8, no. 1, pp. 1–12, 2017.