Learning from Complementary Labels via Partial-Output Consistency Regularization

Deng-Bao Wang,[1,2] **Lei Feng,**[3] **Min-Ling Zhang**[1,2*]
[1]School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
[2]Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China
[3]College of Computer Science, Chongqing University, Chongqing, 400044, China
wangdb@seu.edu.cn, lfeng@cqu.edu.cn, zhangml@seu.edu.cn

Abstract

In *complementary-label learning* (CLL), a multi-class classifier is learned from training instances each associated with complementary labels, which specify the classes that the instance does *not* belong to. Previous studies focus on unbiased risk estimator or surrogate loss while neglect the importance of regularization in training phase. In this paper, we give the first attempt to leverage regularization techniques for CLL. By decoupling a label vector into complementary labels and partial unknown labels, we simultaneously inhibit the outputs of complementary labels with a complementary loss and penalize the sensitivity of the classifier on the partial outputs of these unknown classes by consistency regularization. Then we unify the complementary loss and consistency loss together by a specially designed dynamic weighting factor. We conduct a series of experiments showing that the proposed method achieves highly competitive performance in CLL.

1 Introduction

Multi-class classification has been successfully applied in various real-world applications such as computer vision, natural language processing, and web advertising. However, collecting large-scale accurately labeled data is expensive and thus a critical bottleneck in many tasks. To mitigate this problem, various weakly supervised learning frameworks have been studied in recent years, including *semi-supervised learning* (SSL) [Chapelle *et al.*, 2009; Miyato *et al.*, 2018; Niu *et al.*, 2013; Zhu and Goldberg, 2009], *noisy-label learning* [Han *et al.*, 2018; Feng *et al.*, 2020b; Shu *et al.*, 2019; Li *et al.*, 2020], *positive-unlabeled learning* [Du Plessis *et al.*, 2014; Kiryo *et al.*, 2017], *partial label learning* [Cour *et al.*, 2011; Zhang and Yu, 2015; Zhang *et al.*, 2017; Feng and An, 2019] and *learning from pairwise similarity data* [Bao *et al.*, 2018; Hsu *et al.*, 2018].

Complementary-label learning (CLL) [Ishida *et al.*, 2017; Gao and Zhang, 2021] is a recently proposed weakly supervised learning framework where a multi-class classifier is learned from training instances each associated with complementary labels, which specify the classes that the instance does *not* belong to. In practice, asking a labeler for selecting the correct class from the list of all candidate classes is usually highly time-consuming and even impossible when the number of classes is large. Fortunately, one may choose one of the classes randomly and ask labelers whether a pattern belongs to the chosen class or not, and such a yes/no question is obviously easier to be answered. Another potential application would be data privacy. In some scenarios, collecting data requires asking extremely private questions from users. This procedure may be difficult or return unreliable answers because their privacy considerations. Nonetheless, it would be less mentally demanding if we ask the respondent to provide some incorrect answers or explain that we will transform their provided true label to a complementary label before the data is saved into cloud. Recently, CLL has also been applied to online learning [Kaneko *et al.*, 2019], semi-supervised learning [Chen *et al.*, 2020], and medical image segmentation [Rezaei *et al.*, 2020].

Previous studies of CLL usually focus on unbiased risk estimator and surrogate loss. Ishida *et al.* [2017] and Feng *et al.* [2020a] show that the ordinary classification risk can be recovered by their proposed unbiased risk estimator from only complementarily labeled samples. They also give theoretical results with statistical consistency guarantees. Yu *et al.* [2018] propose a loss correction method with the help of complementary label transition matrix. Ishida *et al.* [2019] derive a framework with an unbiased risk estimator of the classification risk for arbitrary losses and models, and further improve their method by non-negative correction and a gradient ascent trick. Chou *et al.* [2020] propose a surrogate complementary loss framework, which avoids the extremely noisy gradient problem encountered in unbiased risk estimator. However, all previous studies neglect the power of regularization in the training of neural networks. In particular, current state-of-the-art CLL method [Chou *et al.*, 2020] achieves the accuracy of 79.82% on CIFAR-10 with one complementary label per instance. As a counterpart, current SSL method [Xie *et al.*, 2020] that well exploits regularization techniques, achieves highly competitive results, in which they use only 4000 labeled instances, compared with supervised learning.

This paper gives the first attempt to leverage regularization

[*]Corresponding author

techniques to learn with only complementary labels. Focusing on this general purpose, we propose a novel method based on consistency training, namely *partial-output consistency regularization*. By decoupling a label vector into complementary labels and partial unknown labels, we simultaneously inhibit the outputs of these complementary labels with a complementary loss and penalize the sensitivity of the classifier on the partial outputs of these unknown classes by consistency regularization [Tarvainen and Valpola, 2017; Miyato *et al.*, 2018]. Then we unify the complementary loss and consistency loss together by a specially designed dynamic weighting factor. Our method can be easily extended to tackle both the problems of learning from single and multiple complementary labels [Feng *et al.*, 2020a]. A series of experiments demonstrate that our method achieves a new state-of-the-art. For example, our method achieves 94.29% accuracy on CIFAR-10 with only one complementary label per instance, which obtains substantial gain over previous state-of-the-art of 79.82%. Furthermore, our empirical study shows that with the help of consistency regularization, the proposed method obtains comparable performance with the state-of-the-art SSL framework [Xie *et al.*, 2020].

2 Preliminaries

In this section, we first formalize some notations for ordinary multi-class classification, then introduce the preliminary knowledge for learning from complementary labels.

2.1 Ordinary Multi-Class Classification

Let $\mathcal{X} \in \mathbb{R}^d$ denote the feature space with d dimensions, $\mathcal{Y} = \{1, 2, ..., c\}$ denotes the label space with c classes. The labeled sample $(\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y}$ is sampled from an unknown distribution $p(\boldsymbol{x}, y)$. The goal of ordinary multi-class classification is to learn a parameterized function $f(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}^c$ that minimizes the classification risk:

$$\mathcal{R}(f) = \mathbb{E}_{p(\boldsymbol{x},y)} \mathcal{L}\left(f(\boldsymbol{x}), y\right) \tag{1}$$

where $\mathcal{L} : \mathbb{R}^c \times \mathcal{Y} \to \mathbb{R}$ is a multi-class classification loss function. In this paper, we consider a common case where the function f is a deep neural network with the softmax output layer. Since the distribution $p(\boldsymbol{x}, y)$ is unknown, we use the empirical risk $\widehat{\mathcal{R}}(f)$ to approximate $\mathcal{R}(f)$. Assuming a dataset $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ is independently drawn from distribution $p(\boldsymbol{x}, y)$, then we have

$$\widehat{\mathcal{R}}(f) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}\left(f\left(\boldsymbol{x}_i\right), y_i\right) \tag{2}$$

2.2 Complementary-Label Learning

Different from ordinary multi-class classification, in CLL, we are given only a complementary label $\bar{y}$ for each instance. Ishida *et al.* [2017] assume that a complementarily labeled dataset $\{(\boldsymbol{x}_i, \bar{y}_i)\}_{i=1}^n$ is sampled from the following distribution:

$$\bar{p}(\boldsymbol{x}, \bar{y}) = \frac{1}{c-1} \sum\nolimits_{y \neq \bar{y}} p(\boldsymbol{x}, y) \tag{3}$$

which implies that all other labels except the correct label are chosen to be the complementary label with uniform probabilities.

In this paper, we consider a more general setting where each instance is associated with multiple complementary labels, namely *multiple complementary-label learning* (MCLL) [Feng *et al.*, 2020a]. Suppose a MCLL dataset is represented as $\left\{\left(\boldsymbol{x}_i, \bar{Y}_i\right)\right\}_{i=1}^n$, where $\bar{Y}_i$ is a set of complementary labels for the instance $\boldsymbol{x}_i$. Let us denote the size of the complementary label set by a random variable s, which is sampled from a distribution $p(s)$. Then, we assume that each sample is drawn from the following distribution:

$$\bar{p}(\boldsymbol{x}, \bar{Y}) = \sum\nolimits_{j=1}^{c-1} \bar{p}(\boldsymbol{x}, \bar{Y} | s = j) p(s = j), \tag{4}$$

where

$$\bar{p}(\boldsymbol{x}, \bar{Y} | s = j) := \begin{cases} \frac{1}{\binom{c-1}{j}} \sum_{y \notin \bar{Y}} p(\boldsymbol{x}, y), & if |\bar{Y}| = j, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

It is obvious that when $p(s{=}1){=}1$, the distribution expressed by Eq.(4) reduces to the distribution in Eq.(3), hence the MCLL problem results in the single CLL problem. Our goal is still to learn a classifier that minimizes the classification risk (1), but only from complementarily labeled training samples. The empirical risk becomes:

$$\widehat{\mathcal{R}}_{comp}(f) = \frac{1}{n} \sum_{i=1}^{n} \bar{\mathcal{L}}\left(f\left(\boldsymbol{x}_i\right), \bar{Y}_i\right) \tag{6}$$

where $\bar{\mathcal{L}}$ is a specifically designed loss function for learning form complementary labels.

Feng *et al.* [2020a] present a loss formulation which gives unbiased risk estimator by employing any multi-class classification loss $\mathcal{L}$:

$$\begin{aligned} \bar{\mathcal{L}}_{ure}\left(f\left(\boldsymbol{x}_i\right), \bar{Y}_i\right) = & \sum_{y \notin \bar{Y}_i} \mathcal{L}(f(\boldsymbol{x}_i), y) \\ & - \frac{c - 1 - |\bar{Y}_i|}{|\bar{Y}_i|} \sum_{y' \in \bar{Y}_i} \mathcal{L}(f(\boldsymbol{x}_i), y') \end{aligned} \tag{7}$$

When $|\bar{Y}_i| = 1$, which means $p(s = 1) = 1$, the above loss function can be directly used in single CLL. As discussed in [Feng *et al.*, 2020a], this loss formulation is a generalization of [Ishida *et al.*, 2017] and [Ishida *et al.*, 2019]. Chou *et al.* [2020] propose a surrogate complementary loss framework, which achieves highly competitive empirical results, although their proposed surrogate losses are biased to the ordinary classification risk. Different from commonly used losses which are non-increasing functions of the output of the true class, surrogate complementary losses need to be non-decreasing functions of the output on the complementary classes. For example, a modified log loss which minimizes the output of the complementary class is one of the baseline surrogate complementary losses proposed in Chou *et al.* [2020]:

$$\bar{\mathcal{L}}_{scl \cdot log}\left(f\left(\boldsymbol{x}_i\right), \bar{y}_i\right) = -\log(1 - f_{\bar{y}_i}(\boldsymbol{x}_i)) \tag{8}$$

where $f_y(\boldsymbol{x})$ denotes the model output of $\boldsymbol{x}$ on class y. According to the results of our re-implementation, this simply modified log loss achieves current state-of-the-art performance in single CLL problem.

3 Methodology

In this section, we give the first attempt to leverage regularization techniques in CLL. We present the overview of our proposed method in Subsection 3.1, then introduce the detailed techniques in Subsection 3.2 and 3.3. We discuss a further extension of CLL framework in Subsection 3.4.

3.1 Partial-Output Consistency Regularization

A recent line of work in weakly supervised learning has been proposed to utilize unlabeled samples to enforce smoothness and consistency of the model predictions [Tarvainen and Valpola, 2017; Miyato *et al.*, 2018; Xie *et al.*, 2020]. Generally speaking, their proposed methods simply regularize model prediction outputs to be invariant to small changes applied to the input space of samples. The core idea of this framework is intuitive because a good model should be robust to small perturbations in the input space of samples.

Following this idea, we propose to use consistency regularization on partial outputs to learn from complementary labels. Firstly, we decouple the label vector of each sample into two parts: complementary labels and partial unknown labels. On these complementary labels, we can minimize the corresponding conditional probabilities to inhibit the outputs of the learned model. In recent years, several surrogate complementary losses have been proposed to this end [Chou *et al.*, 2020; Feng *et al.*, 2020a]. For example, the complementary log loss (see in Eq.(8)) can be used in single CLL scenario, and we can extend it to a more generic formulation which can learn from both single and multiple complementary labels:

$$\bar{\mathcal{L}}_{log}\left(f\left(\boldsymbol{x}_i\right), \bar{Y}_i\right) = -\sum\nolimits_{y\in\bar{Y}_i}\log(1-f_y(\boldsymbol{x}_i)) \tag{9}$$

Then we employ consistency regularization on the partial unknown classes to penalize the sensitivity of the classifier outputs among inputs with small noises. According to the basic assumption that the outputs of a robust model should not be significantly affected by natural and small input changes, one can improve the robustness of the model by minimizing a divergence metric between the outputs of the original sample and samples with injected small noises. Xie *et al.* [2020] suggest that data augmentation methods, which are widely used for expanding training data size in supervised learning, can lead to strong performance when used in consistency training framework. These advanced data augmentation techniques can preserve the label of the original example while are diverse and natural. Formally, the objective of partial-output consistency regularization can be expressed as the following cross-entropy between the prediction of original input and augmentations:

$$\begin{aligned}\bar{\mathcal{L}}_{consist}\big(f(\boldsymbol{x}_i), \bar{Y}_i\big) = \\ -\sum_{j=1}^{m}\sum\nolimits_{y\notin\bar{Y}_i}\widetilde{f}_y(\boldsymbol{x}_i)\log\big(f_y(\mathcal{AUG}_j(\boldsymbol{x}_i))\big)\end{aligned} \tag{10}$$

where $\mathcal{AUG}_j(\boldsymbol{x}_i)$ denotes the j-th augmented version of original input $\boldsymbol{x}_i$, and m is the number of augmentations used for consistency training. The practical implementation of this augmentation process will be introduced in next subsection. $\widetilde{f}_y(\boldsymbol{x}_i)$ denotes the re-normalized output of $\boldsymbol{x}_i$ on class y, which is difined as:

$$\widetilde{f}_y(\boldsymbol{x}_i) = \begin{cases}\frac{\hat{f}_y(\boldsymbol{x}_i)}{\sum_{k\notin\bar{Y}_i}\hat{f}_k(\boldsymbol{x}_i)}, & if\ y\notin\bar{Y}_i, \\ 0, & \text{otherwise.}\end{cases} \tag{11}$$

The re-normalized process makes sure that the distribution of partial outputs on unknown classes is still a valid probability distribution. Note that we use $\hat{f}$, a fixed copy of the current model, to detach the gradient propagation through $\widetilde{f}(\boldsymbol{x}_i)$, as suggested by Miyato *et al.* [2018]. Then we can unify the complementary loss and consistency regularization loss together under a single loss function, which is defined as:

$$\begin{aligned}\bar{\mathcal{L}}_{total}\left(f\left(\boldsymbol{x}_i\right), \bar{Y}_i\right) = \\ \bar{\mathcal{L}}_{log}\left(f\left(\boldsymbol{x}_i\right), \bar{Y}_i\right) + \lambda\bar{\mathcal{L}}_{consist}\big(f(\boldsymbol{x}_i), \bar{Y}_i\big)\end{aligned} \tag{12}$$

where the complementary loss $\bar{\mathcal{L}}_{log}$ can be replaced by other losses like exponential loss or upper-bound log/exponential loss [Feng *et al.*, 2020a]. The weighting factor λ is used to balance the complementary loss and the consistency loss, which will be shown important for stable training in Subsection 3.3. Finally, we replace $\bar{\mathcal{L}}$ in Eq.(6) by the unified loss $\bar{\mathcal{L}}_{total}$ and learn by minimizing the empirical risk with $\bar{\mathcal{L}}_{total}$.

3.2 Data Augmentation

The augmentations used in Eq.(10) need to be different from the original input as well as preserve the semantic information. In recent years, different augmentation strategies have been proposed to generate diverse and natural augmentations. Among existing data augmentation methods, AutoAugment [Cubuk *et al.*, 2019] is an empirically promising method which automatically searches for augmentation policies from PIL, a popular Python image library, by evaluating the quality of a particular policy directly on the dataset of interest. In this work, we use the searched policies released by Cubuk *et al.* [2019] in our consistency training procedure. Following the original literature of AutoAugment, we concatenate the best searched policies into a policy pool, and randomly choose a policy to produce an augmentation. We additionally use another promising augmentation technique Cutout [DeVries and Taylor, 2017] after applying AutoAugment. Although we restrict our attention to the image classification tasks in this paper, it is worth noting that for language tasks, back-translation has been successfully used to generate augmentations which are diverse while preserving the semantics of the original sentences [Edunov *et al.*, 2018; Cubuk *et al.*, 2019].

3.3 Dynamic Weighting Factor

The overall objective presented in Subsection 3.1 is a weighted combination of the complementary loss and the consistency loss, controlled by a hyperparameter λ. Empirically, we observe that the weighting factor λ significantly affects the performance of our method. We use an illustrative experiment to demonstrate this point.

We adopt PreAct-ResNet-18 network [He *et al.*, 2016] as the base model and train on CIFAR-10 dataset with fixed λ equal to 0.1 and 1 respectively (the implementation details are the same with Subsection 4.1). Figure 1 (left) shows

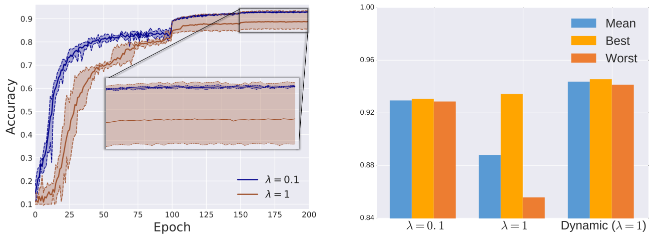


Figure 1: Test accuracies of PreAct-ResNet-18 network on CIFAR-10 with different weighting factors. Left: Test accuracy curves with λ equals to 0.1 and 1 respectively. Solid lines show the mean accuracies of 5 trials and dashed lines show the best and worst accuracies among 5 trials. Right: Comparison results with different weighting schemes.

the mean, best and worst of test accuracies of 5 trials. We can see that: (1) when using a smaller weighting factor, which means the unified loss is dominated by complementary loss, model always achieves better performance in the initial training stage; (2) when using a larger weighting factor, the discrepancy of test accuracies between different trials becomes significantly larger, and the mean accuracy is much lower than that trained with smaller weighting factor; (3) the best performance among 5 trials with $\lambda = 1$ is better than that with $\lambda = 0.1$ in the end of training even if it is worse than all cases with $\lambda = 0.1$ in the beginning epochs.

This phenomenon may be caused by the low quality model outputs in early stage of training. Specifically, in the beginning epochs, model may produce highly random predictions, thus the consistency loss which involves those low quality predictions would cause error accumulation problem during training. This suggests us to use a small weighting factor in the beginning of training and relatively large one in the later stage. To this end, we propose a dynamic weighting scheme which gradually increases this factor during training epochs. Specifically, we use an increasing function to obtain this dynamic factor at epoch t:

$$\Lambda(t) = \min\{\frac{t}{T'}\lambda, \lambda\} \tag{13}$$

After substituting the fixed weighting factor λ in Eq.(12) by above function, our method trains with weighting factor which equals to 0 at the beginning epoch and increases it to λ at epoch T'. After T' epochs, it keeps a constant λ until the end of training. We also use an illustrative comparison between our dynamic weighting scheme and constant weighting scheme to demonstrate the effectiveness of the dynamic scheme. As is shown in Figure 1 (right), the dynamic weighting scheme achieves high mean accuracy as well as stable performance across all trials.

3.4 Further Extension

In some particular situations, we may have ordinarily labeled data in addition to complementarily labeled data. Practically, asking a labeler for selecting the true label from all candidates is usually difficult and even impossible in some domains. Instead, the yes/no question which asks whether an instance belongs to a specific class is easier to be answered. Based on this query type, one can choose to iteratively ask labelers about an instance until its true label is recovered, which results in a SSL problem. One can also query each instance with a single time though the true labels may not be recovered. The latter strategy gives a dataset with two parts: ordinarily labeled subset and complementarily labeled subset. This particular learning setting, i.e. learning from both ordinary and complementary labels, can be considered as an alternative to SSL, and our consistency training framework can be directly used in this problem by transiting an ordinary label to $c - 1$ complementary labels. We will use a particular experiment to demonstrate the superiority of our method compared with the SSL framework under same data labeling cost in experiment section.

4 Experiments

4.1 Experimental Setup

Datasets

To verify the superiority of our method, we conduct experiments on three commonly used image datasets: SVHN, CIFAR-10 and CIFAR-100. SVHN contains 73,257 training samples and 26,032 test samples. Both of CIFAR-10 and CIFAR-100 contain 50,000 training samples and 10,000 test samples. SVHN and CIFAR-10 have 10 classes and CIFAR-100 has 100 classes. We conduct experiments by considering both the scenarios of single CLL and MCLL. To generate single complementary label, we randomly select one of the complementary classes per instance. To generate multiple complementary labels, we first instantiate $p(s) = \binom{c-1}{s}/(2^c - 2), \forall s \in \{1, ..., c-1\}$, which represents the ratio of the number of label sets whose size is s to the number of all possible label sets. Then for each instance $\boldsymbol{x}$, we sample s from $p(s)$, and then sample a complementary label set $\bar{Y}$ with size s from $p(\bar{Y}) = 1/\binom{c-1}{s}$.

Comparison methods

For single CLL, we compare our method with PC [Ishida *et al.*, 2017], GA [Ishida *et al.*, 2019], Forward [Yu *et al.*, 2018], SCL-EXP and SCL-LOG [Chou *et al.*, 2020], where SCL means *surrogate complementary loss*. We also compare with two MCLL methods: UB-EXP and UB-LOG [Feng *et al.*, 2020a], where UB is the short form of *upper bounded*, since single CLL can be considered as a special case of MCLL. For MCLL, except these two native MCLL methods, we also compare with SCL-EXP and SCL-LOG by extending them to learn from multiple complementary labels (see in Eq.(9)). Note that PC, GA, and Forward can also be used in MCLL after decomposing each sample into multiple samples each with a single complementary label. However, the empirical results of [Feng *et al.*, 2020a] showed that UB-EXP and UB-LOG consistently outperform the decomposition strategy. Thus the decomposition-based methods are not involved in the MCLL scenario.

Implementation

Our implementation is based on PyTorch [Paszke *et al.*, 2019] and experiments were carried out with NVIDIA Tesla V100 GPU. We train the commonly used LeNet-5, PreAct-ResNet-18 and Wide-ResNet-34-10 with 200 epochs, and use SGD

Table 1: Comparison of classification accuracies between different methods using different network architectures on SVHN and CIFAR-10 with one complementary label per instance. The results (mean±std) are reported over 5 random trials.

Dataset	SVHN		CIFAR10	
Model	LeNet-5	PreAct-ResNet-18	PreAct-ResNet-18	Wide-ResNet-34-10
PC [Ishida *et al.*, 2017]	32.13±0.82%	27.16±0.34%	36.97±0.86%	35.01±0.42%
Forward [Yu *et al.*, 2018]	84.51±0.28%	90.83±0.04%	77.22±0.84%	78.69±0.54%
GA [Ishida *et al.*, 2019]	59.75±2.14%	57.83±0.32%	42.19±2.09%	42.32±0.04%
UB-EXP [Feng *et al.*, 2020a]	84.02±0.07%	89.73±0.04%	75.00±1.43%	77.86±0.99%
UB-LOG [Feng *et al.*, 2020a]	83.48±0.62%	90.10±0.10%	75.02±1.56%	77.29±1.10%
SCL-EXP [Chou *et al.*, 2020]	83.63±0.64%	89.45±0.46%	77.51±0.33%	79.17±0.50%
SCL-LOG [Chou *et al.*, 2020]	84.45±0.37%	90.80±0.28%	79.82±0.43%	81.55±0.28%
Ours	**93.12±0.23%**	**96.69±0.04%**	**94.29±0.16%**	**95.72±0.15%**

Table 2: Comparison of classification accuracies (%) between different methods on SVHN, CIFAR-10 and CIFAR-100 with multiple complementary labels associated with each instance. A&C means AutoAugment and Cutout.

Dataset	SVHN	CIFAR10	CIFAR100
SCL-EXP	90.46±0.11	91.99±0.17	45.36±1.12
SCL-LOG	90.27±0.12	92.86±0.08	46.82±0.64
UB-EXP	90.27±0.24	91.27±0.10	28.03±1.42
UB-LOG	89.90±0.25	92.51±0.11	47.92±2.62
UB-LOG with A&C	94.42±0.01	93.57±0.16	21.94±0.93
Ours w/o Re-norm	**95.03±0.07**	**96.07±0.20**	50.49±0.18
Ours	**95.01±0.09**	**96.09±0.15**	**54.17±0.89**

as the opimizer with a momentum of 0.9, a weight decay of 1e-4, and a batch size 64 in our experiments. We set the initial learning rate as 0.1 across all datasets and divide it by a factor of 10 after 100 epochs and 150 epochs respectively. The hyperparameters used in Eq.(13) are set as $T' = 100$ and $\lambda = 1$. The number m of augmented instances used for consistency training is set to 2 and these augmentations are generated using the techniques discussed in Subsection 3.2. For CIFAR-10 and CIFAR-100 we further use three standard image pre-processing techniques, normalization, horizontal flipping and random cropping, to all training samples. In single CLL experiments, we use SCL-LOG as the complementary loss as is shown in Eq.(12). In MCLL experiments, we replace the complementary loss SCL-LOG in Eq.(12) by UB-LOG, which is specifically designed for MCLL.

We re-implement all the comparison methods using the same baseline image pre-processing techniques, network architecture, optimizer and learning policies with our method except PC and GA, in which we failed to achieve comparable results using our policies. Hence we maintain the learning policies reported in their original literatures.

4.2 Experimental Results

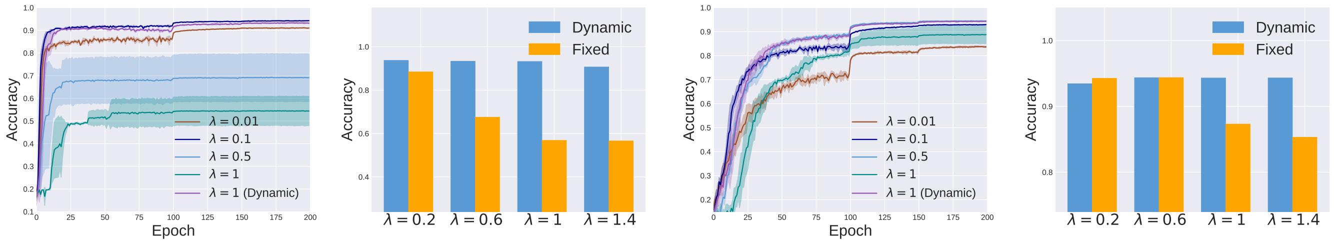
Comparison results

Table 1 presents the comparison results on SVHN and CIFAR-10 in single CLL scenario. We use different network architectures according to the complexity of each dataset. In our experiments, the baseline image pre-processing techniques are adopted in all methods for fair comparison, thus our re-implementation results are better than the original results reported in their literatures. Nevertheless, our method achieves new state-of-the-art results across all cases. On SVHN, which is a relatively simple and easy dataset, existing methods can already achieve considerably high accuracies. With the help of consistency training, we push the state-of-the-art one step forward. The improvements of performance are more significant on CIFAR-10, which is a more complex dataset than SVHN. In particular, when using PreAct-ResNet-18, our method achieves the accuracy of 94.29%, which is significantly better than the previous state-of-the-art accuracy of 79.82% by SCL-LOG. Additionally, by learning with a single complementary label per instance, our method obtains surprisingly comparable results with ordinary supervised learning (see in Table 3).

Table 2 presents the comparison results on SVHN, CIFAR-10 and CIFAR-100 in MCLL scenario. Here we use LeNet-5 on SVHN and PreAct-ResNet-18 on CIFAR datasets. UB-LOG with A&C means we additionally apply AutoAugment and Cutout after the baseline image pre-processing when learning with UB-LOG. We also compare with a degenerate version of our method, in which the re-normalization process (i.e. Eq.(11)) is abandoned. In MCLL experiments, we use the complementary label generation strategy introduced in Subsection 4.1, thus the expected number of complementary labels per instance is $c/2$. As shown in Table 2, our method still outperforms all baselines in MCLL scenario. In addition, the results of UB-LOG with A&C show that simply using these augmentations in CLL can not give improvements in all cases.

Parameter sensitivity analysis

In Figure 2, we show the learning curves of dynamic weighting scheme and constant weighting scheme with different factor values. The results are very similar with the illustrative experiment presented in Subsection 3.3. The model tends to have unstable performance when adopting larger weighting factors in the initial stage. This issue is especially serious on SVHN, on which the mean result with $\lambda = 1$ is significantly incomparable. When using constant weighting scheme, the



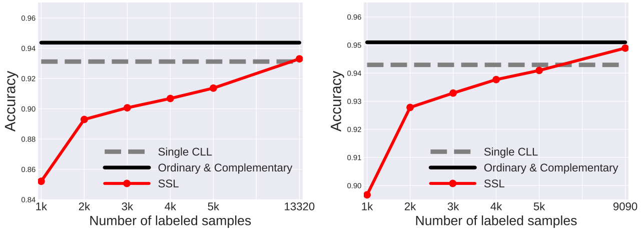
(a) SVHN

(b) CIFAR-10

Figure 2: Test accuracy curves with fixed λ as well as dynamic weighting strategy, and comparison results between fixed and dynamic weighting strategies with different λ. Solid curves show the mean accuracies of 5 trials and light colors show the standard deviation. The experiments are conducted with LeNet-5 and PreAct-ResNet-18 on SVHN and CIFAR-10 in single CLL scenario.

Method	PreAct-ResNet-18	Wide-ResNet-34-10
[Yu *et al.*, 2018]	77.22%	78.69%
[Feng *et al.*, 2020a]	75.02%	77.29%
[Chou *et al.*, 2020]	79.82%	81.55%
Ours	94.29%	95.72%
SSL with 4k Labels	93.77%	95.09%
Full Supervision	96.16%	97.17%

Table 3: Classification accuracy of different methods on CIFAR-10. Complementary-label learning methods (top) are learned with one complementary label per instance.



(a) SVHN

(b) CIFAR-10

Figure 3: Comparison results between our method and the SSL framework. We use LeNet-5 and PreAct-ResNet-18 on SVHN and CIFAR010 respectively.

preferable values of λ on SVHN and CIFAR-10 vary widely, which means this value needs to be manually searched from a validation set in practice. Fortunately, the proposed dynamic weighting scheme can effectively alleviate this issue. When adopting dynamic weighting scheme with $\lambda = 1$, we can see that the mean accuracy is very close to the best one of constant weighting factors on both datasets. For reliable usage, the hyperparameter λ in our dynamic weighting scheme can be chosen around 1. We also show the comparison results between the constant and dynamic weighting schemes with λ ranging form 0.2 to 1.4, the comparison results demonstrate that the dynamic weighting scheme is more robust to λ.

Comparison with the SSL framework

As we discussed in Subsection 3.4, in some particular domains, CLL can be considered as an alternative to SSL. To empirically verify this point, we re-implement the consistency training-based SSL framework [Xie *et al.*, 2020] with the same network, learning policies, and data augmentations of our method. Figure 3 presents the results of our method trained in single CLL (gray line) and the SSL framework with controlled number of labels (red line). In addition, we consider a practical scenario where we can only query instance-label pairs and obtain the corresponding yes/no answers. Suppose the total query times $q = m$, where m denotes the size of dataset. By randomly choosing one label per instance to ask the labeler, we can obtain a dataset with m/c ordinarily labeled samples and $m(c-1)/c$ complementarily labeled samples. Besides, we can iteratively query the labels for each instance until its true label is recovered, which results in $2m/(c+1)$ labeled samples and $m(c-1)/(c+1)$ unlabeled samples. From this perspective, we conduct experiments to compare our CLL method with the SSL framework under same data labeling cost. For SVHN, we use our method with 7,325 ordinarily labeled samples in addition to 65,932 complementarily labeled samples, and use the SSL framework using 13,320 labeled samples in addition to 59,937 unlabeled samples. For CIFAR-10, we use our method with 5,000 ordinarily labeled samples as well as 45,000 complementarily labeled ones, and use 9,090 labeled samples in addition to 40,910 unlabeled ones in the SSL framework. As shown in Figure 3, in this special case, our method (black line) obtains highly competitive performance compared with the SSL framework.

5 Conclusion

In this paper ,we give the first attempt to leverage the consistency training framework in CLL. We present a unified loss which simultaneously inhibits the outputs of complementary classes and penalizes the output sensitivity on the partial unknown classes. Based on the empirical observation of the importance of weighting factor, we propose a dynamic weighting scheme which helps our consistency training framework obtain stable and accurate results. We conduct a series of experiments showing that the proposed method achieves a new state-of-the-art in CLL. Furthermore, we compare our method with state-of-the-art SSL framework under same labeling cost, which demonstrates that our method is highly competitive compared with SLL framework.

References

Han Bao, Gang Niu, and Masashi Sugiyama. Classification from pairwise similarity and unlabeled data. In *ICML*, pages 452–461, 2018.

Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

John Chen, Vatsal Shah, and Anastasios Kyrillidis. Negative sampling in semi-supervised learning. In *ICML*, pages 1704–1714, 2020.

Yu-Ting Chou, Gang Niu, Hsuan-Tien Lin, and Masashi Sugiyama. Unbiased risk estimators can mislead: A case study of learning with complementary labels. In *ICML*, pages 1929–1938, 2020.

Timothee Cour, Ben Sapp, and Ben Taskar. Learning from partial labels. *The Journal of Machine Learning Research*, 12:1501–1536, 2011.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *CVPR*, pages 113–123, 2019.

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv:1708.04552*, 2017.

Marthinus C Du Plessis, Gang Niu, and Masashi Sugiyama. Analysis of learning from positive and unlabeled data. *NeurIPS*, pages 703–711, 2014.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale. In *EMNLP*, pages 489–500, 2018.

Lei Feng and Bo An. Partial label learning with self-guided retraining. In *AAAI*, pages 3542–3549, 2019.

Lei Feng, Takuo Kaneko, Bo Han, Gang Niu, Bo An, and Masashi Sugiyama. Learning with multiple complementary labels. In *ICML*, pages 3072–3081, 2020.

Lei Feng, Senlin Shu, Zhuoyi Lin, Fengmao Lv, Li Li, and Bo An. Can cross entropy loss be robust to label noise? In *IJCAI*, pages 2206–2212, 2020.

Yi Gao and Min-Ling Zhang. Discriminative complementary-label learning with weighted loss. In *ICML*, 2021. in press.

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. *NeurIPS*, pages 8527–8537, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, pages 630–645, 2016.

Yen-Chang Hsu, Zhaoyang Lv, Joel Schlosser, Phillip Odom, and Zsolt Kira. Multi-class classification without multi-class labels. In *ICLR*, 2018.

Takashi Ishida, Gang Niu, Weihua Hu, and Masashi Sugiyama. Learning from complementary labels. In *NeurIPS*, pages 5639–5649, 2017.

Takashi Ishida, Gang Niu, Aditya Menon, and Masashi Sugiyama. Complementary-label learning for arbitrary losses and models. In *ICML*, pages 2971–2980, 2019.

Takuo Kaneko, Issei Sato, and Masashi Sugiyama. Online multiclass classification based on prediction margin for partial feedback. *arXiv:1902.01056*, 2019.

Ryuichi Kiryo, Gang Niu, Marthinus C Du Plessis, and Masashi Sugiyama. Positive-unlabeled learning with non-negative risk estimator. In *NeurIPS*, pages 1675–1685, 2017.

Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *ICLR*, Addis Ababa, Ethiopia, 2020.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993, 2018.

Gang Niu, Wittawat Jitkrittum, Bo Dai, Hirotaka Hachiya, and Masashi Sugiyama. Squared-loss mutual information regularization: A novel information-theoretic approach to semi-supervised learning. In *ICML*, pages 10–18, 2013.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8026–8037, 2019.

Mina Rezaei, Haojin Yang, and Christoph Meinel. Recurrent generative adversarial network for learning imbalanced medical image semantic segmentation. *Multimedia Tools and Applications*, 79(21):15329–15348, 2020.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *NeurIPS*, pages 1919–1930, 2019.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, pages 1195–1204, 2017.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *NeurIPS*, pages 6256–6268, 2020.

Xiyu Yu, Tongliang Liu, Mingming Gong, and Dacheng Tao. Learning with biased complementary labels. In *ECCV*, pages 68–83, 2018.

Min-Ling Zhang and Fei Yu. Solving the partial label learning problem: an instance-based approach. In *IJCAI*, pages 4048–4054, 2015.

Min-Ling Zhang, Fei Yu, and Cai-Zhi Tang. Disambiguation-free partial label learning. *IEEE Transactions on Knowledge and Data Engineering*, 29(10):2155–2167, 2017.

Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 3(1):1–130, 2009.