Balanced and Accurate Pseudo-Labels for Semi-Supervised Image Classification

JIAN ZHAO, XIANHUI LIU, and WEIDONG ZHAO, Tongji University

Image classification by semi-supervised learning has recently become a hot spot, and the Co-Training framework is an important method of semi-supervised image classification. In the traditional Co-Training structure, the sub-networks will generate pseudo-labels for each other, and these pseudo-labels will further be used as a supervisory signal for model training. However, the pseudo-labels will hurt classification performance because of their low accuracy and unbalanced distribution. In this article, we are trying to solve the preceding two problems by designing the Balanced Module (BM) and Gaussian Mixture Module (GMM), and propose BAPS (the *B*alanced and *A*ccurate *P*seudo-labels for *S*emi-supervised image classification). In BM, the two sub-networks jointly predict the unlabeled images, then select the pseudo-labels with a high-confidence threshold to perform the balancing operation to obtain the initial samples with balanced distribution of each category. In GMM, referring to the common practice of the Learning from Noise Labels task, we use GMM to fit the loss distribution of images with pseudo-labels output by BM, then clean samples and noise samples are divided based on the observation that the loss of correctly labeled images is generally smaller than that of wrongly labeled ones. Through BM and GMM, pseudo-labels with balanced distribution and high accuracy are obtained for the subsequent model training process. Our model has achieved better classification accuracy than most state-of-the-art semi-supervised image classification algorithms on the CIFAR-10/100 and SVHN datasets, and further ablation experiments demonstrate the effectiveness of our BAPS. The source code of BAPS will be available at https://github.com/zhaojianaaa.

CCS Concepts: • **Computing methodologies** → **Semi-supervised learning settings**; **Image representations**;

Additional Key Words and Phrases: Deep Co-Training, Balanced Module, Gaussian Mixture Module, semi-supervised classification

ACM Reference format:
Jian Zhao, Xianhui Liu, and Weidong Zhao. 2022. Balanced and Accurate Pseudo-Labels for Semi-Supervised Image Classification. *ACM Trans. Multimedia Comput. Commun. Appl.* 18, 3s, Article 145 (October 2022), 18 pages.
https://doi.org/10.1145/3506711

This work was supported by the National High-Technology Research and Development Program of China (no. 2017YFB0304102), the Shanghai Innovation Action Project of Science and Technology (no. 21511104300), and the Fundamental Research Funds for the Central Universities.

Authors' address: J. Zhao, X. Liu (corresponding author), and W. Zhao, Tongji University, No. 4800 Cao'an Road, Jiading District, Shanghai, China 201804; emails: zjtju1919@gmail.com, {1910678, wd}@tongji.edu.cn.

1551-6857/2022/10-ART145 $15.00
https://doi.org/10.1145/3506711

1 INTRODUCTION

With the increasing enthusiasm of deep learning research, various deep learning models [1–5] have been proposed successively. Deep learning has achieved excellent performance in many fields, especially in computer vision, such as image classification [15–18], object detection [6, 7], and semantic segmentation [8, 9].

However, the training process of deep learning requires a large number of labeled samples. In real-world applications, it is extremely expensive to obtain a large number of labeled samples, whereas unlabeled ones are easy to get. Therefore, the difficulty lies in how to use unlabeled samples. Semi-supervised learning algorithms trained on a small number of labeled samples and a large number of unlabeled samples are more suitable for real-world application scenarios. In recent years, it has become a popular research direction in the field of deep learning.

There are four approaches for image classification based on deep semi-supervised learning: Co-Training [15–18], Consistency Regularization [19–22], Mixup [23–25], and **semi-supervised Generative Adversarial Network (SGAN)** [26–28]. The Co-Training method usually uses more than one network for collaborative training. First, the small number of labeled images will be used to construct a labeled training dataset for each sub-network in the Co-Training framework. During the training process, one sub-network will predict all remaining unlabeled images and select images with high-confidence pseudo-labels as the other sub-network's new training data. Sub-networks will use these images with pseudo-labels to conduct supervised learning.

This Co-Training framework has two serious problems. First, some pseudo-labels generated by the sub-networks are wrong, and the incorrectly labeled images cannot be revised in the subsequent training process, which will make the model overfit the noise samples, resulting in poor generalization performance [10, 29–32]. Second, this operation of generating pseudo-labels cannot guarantee the balance of the distribution of various categories in the training dataset, which will affect the training of the model [11–14].

To overcome the preceding two problems, we designed the **Balanced Module (BM)** and **Gaussian Mixture Model (GMM)** on the basis of the traditional Co-Training framework that includes two sub-networks, which are used to handle the imbalanced dataset and low accuracy of the pseudo-labels, respectively. In BM, first, the two sub-networks jointly predict all unlabeled images and select the ones with high-confidence pseudo-labels as the initial chosen dataset. Then, by analyzing the category distribution of the initial chosen dataset, we chose the same number of images of each category to form a balanced dataset. However, this balanced dataset contains some incorrectly labeled images, and the proportion of that will be much higher in the later training stage. To better select the correctly labeled images from the pseudo-labeled samples, we learn from the task of **Learning from Noise Labels (LNL)** and introduce GMM. The LNL method tells us that the loss of incorrectly labeled samples is greater than that of truly labeled ones. So we use GMM to fit the loss distribution of the sample output by BM and select the parts with less loss as the clean dataset for model training.

In our BAPS (*B*alanced and *A*ccurate *P*seudo-labels for *S*emi-supervised image classification) model, the two sub-networks are the modified version of FixMatch [22]. FixMatch is a deep semi-supervised learning method based on Mixup, which uses cross-entropy loss for supervised training on labeled samples. For unlabeled images, the prediction of the weakly augmented image are used as the supervisory signal for the corresponding strongly augmented ones for supervised training. Obviously, if the pseudo-labels of the weakly augmented images are false, the model will be trained with noise labels, which will seriously interfere with their learning ability. It seems that this interference could be alleviated to a certain extent by increasing the confidence threshold, but in this way, the samples retained for training will be reduced. In fact, the prediction accuracy of unlabeled images during the training process of FixMatch is about 96%, whereas the two sub-networks in BAPS both reach 99%. Additionally, the distribution of the training dataset obtained by directly applying a confidence threshold on the weakly augmented images is also imbalanced. BM and GMM introduced in this article can obtain a balanced and accurate training dataset. Further, considering the large number of samples divided into noise dataset by GMM, we additionally introduced a **Kullback-Leibler (KL)** loss. Compared with FixMatch, the usage of unlabeled data by KL loss is more reasonable, as KL loss will not bring negative supervisory information due to the wrong predictions but just emphasizes the distribution consistency between the weakly and strongly augmented images.

The contributions of this article are as follows:

- BAPS overcomes the imbalance problem of the training dataset in the traditional Co-Training model. By designing BM, the distribution of each category in the training dataset becomes more balanced, and the training process is more efficient.
- Under the Co-Training framework, drawing lessons from the Co-Teaching framework in the LNL task, GMM is introduced to make the two sub-networks divide clean samples for each other and accurately select the correctly labeled images from all unlabeled datasets with pseudo-labels. In this way, the negative impact of wrongly labeled samples on model training is weakened.
- Extensive quantitative evaluations on CIFAR-10/100 and SVHN real scene image classification datasets demonstrate that our proposed BAPS performs favorably against most state-of-the-art deep semi-supervised image classification methods.

The rest of this article is organized as follows. Section 2 describes the related work. The details of our proposed BAPS are presented in Section 3. Experimental results and discussions are given in Section 4. Section 5 presents the conclusion.

2 RELATED WORK

2.1 Deep Semi-Supervised Image Classification

The semi-supervised image classification methods could be roughly divided into four categories: Co-Training, Consistency Regularization, Mixup, and SGAN.

2.1.1 Co-Training. The Co-Training framework consists of at least two sub-networks, and all sub-networks are trained collaboratively. Blum and Mitchell [15] proposed the first Co-Training model. During the training process, one sub-network predicts unlabeled samples and selects the high-confidence pseudo-labels into the training dataset of the other sub-network. In short, sub-networks will provide pseudo-labels for each other. Qiao et al. [16] applied deep learning structures to Co-Training for the first time and carried out image classification tasks. Unlike traditional Co-Training, it does not construct a different training dataset for each sub-network; instead, considering the view difference constraints, the adversarial example is generated for each image through the **Generative Adversarial Network (GAN)**, and the images with the corresponding adversarial ones are jointly trained to maintain the difference of each sub-network. Chen et al. [17] proposed tri-net, which is a Co-Training framework that consists of three sub-networks. The labeled samples are repeatedly sampled to obtain three different labeled training datasets, which are used to train three different classifiers for image classification. In the collaborative training process, the new training dataset of each classifier is provided by the other two classifiers. Mo et al. [18] improved the accuracy of pseudo-labels in tri-net. By setting a weight for each pseudo-label and further obtaining the confidence threshold of them based on the information entropy, the high-accuracy pseudo-labels are selected as the training dataset.

2.1.2 Consistency Regularization. The Co-Training framework uses unlabeled data by predicting their pseudo-labels for supervised training, whereas Consistency Regularization uses KL divergence or mean square error to minimize the predictions of the same image after different enhancements for unsupervised training. Laine and Aila [19] proposed π-model, which uses cross-entropy loss for labeled images, whereas for unlabeled ones, π-model encourages consistent network outputs between two augmentations of the same image. Since π-model requires two forward propagations to update the model parameters, which slow down the model training process, Tarvainen and Valpola [20] proposed the Teacher Student network on the basis of π-model. Input images were predicted by the Student and Teacher network, respectively, and the consistency loss of the two outputs were used to train the Student network, whereas the model parameters of the Teacher network were updated by the exponential moving average of model parameters from the Student network. Xie et al. [21] proposed UDA, which still calculates cross-entropy loss for labeled images and Consistency Regularization loss for unlabeled images, but compared with the previous simple crop or flip image augmentation methods, UDA designed a more complex approach—RandAugment, which improved the accuracy of image classification. FixMatch proposed by Sohn et al. [22] directly takes the high-confidence predictions of weakly augmented images as a supervisory signal for the corresponding strongly augmented ones to conduct supervised training. FixMatch is very simple and efficient, but if the confidence threshold is not set properly, the model training could be affected.

2.1.3 Mixup. The method based on Consistency Regularization conducts network training by emphasizing the consistency of the same images through different augmentations. The images before and after enhancement are basically the same one, whereas Mixup generates new images by linearly combining the images from the training dataset. Mixup was originally proposed by Zhang et al. [23] and is used for supervised learning algorithms. By adopting Mixup, more new training images are generated, which brings diversity to the training set, and the model becomes more robust. Berthelot et al. [24] introduced Mixup to the field of semi-supervised learning and proposed MixMatch, in which the high-quality pseudo-labels of unlabeled images are obtained by averaging the predictions multiple times. Then, the labeled images and unlabeled images with pseudo-labels are mixed up to generate new images and corresponding labels. Finally, the model is trained by the cross-entropy loss with the labeled images and mean square error with the pseudo-labeled images. Berthelot et al. [25] further proposed ReMixMatch. Considering that it is not reasonable to directly average multiple predictions by different augmentations in MixMatch, ReMixMatch uses the predictions of the weakly augmented images as pseudo-labels of the strongly augmented ones.

2.1.4 Semi-Supervised GAN. GAN is a deep generative model proposed by Goodfellow et al. [26] in 2014. Through the mutual game learning of the generator and the discriminator, we can get accurate outputs. Salimans et al. [27] introduced GAN to semi-supervised learning and proposed SSL-GAN (semi-supervised learning GAN). For the k classification problem, the discriminator is set to be a $k + 1$ classifier, and class $k + 1$ is the fake image generated by the generator. There are three kinds of inputs in the discriminator: labeled images, unlabeled images, and images produced by the generator. For labeled images, cross-entropy loss is used, and for unlabeled images, the discriminator needs to classify them as one of the first k classes, for samples generated by the generator, needs to classify as class $k + 1$. Odena [28] simplified SSL-GAN as SGAN. The inputs of the discriminator in SGAN only contain labeled samples and samples generated by the generator, and the number of samples in the two parts is roughly the same in each batch during training.

2.2 Learning from Noise Labels

The methods of LNL could be roughly divided into three categories. The first approach aims to make the training process more stable to noise samples by modifying the loss function. The second approach directly divides the entire dataset into clean and noise parts, which are used for supervised and unsupervised training, respectively. The third one improves the model's robustness to noisy labels by constructing different network structures.

2.2.1 Methods Based on Improved Loss Functions. In the training process based on the error backpropagation algorithm, the model tends to fit samples with greater loss, and the loss of incorrectly labeled images is obviously greater than that of correctly labeled ones, so traditional cross-entropy loss encourages the model to learn wrong information from noise labels. Based on this idea, Reed et al. [10] revised the cross-entropy loss by linearly superimposing the predictions and training labels, which increases the loss of the clean samples and reduces that of the noise ones, and alleviated the bad effect of noise samples. Park et al. [29] apply the label smoothing operation by injecting uniform noise to the training labels, which increases the model's robustness to noise samples.

2.2.2 Division-Based Methods. The division-based method hopes to find the correctly labeled samples from all pseudo-labeled ones directly. Nguyen et al. [30] observed that predictions of noise labels are more volatile, so they designed the SELF (Self-Ensemble Label Filtering) model, which selects the images with consistent multiple predictions as clean samples for model training. Park et al. [29] proposed confidence-based, metric-based, and hybrid-based methods to extract clean samples. The confidence-based method treats images with high-confidence pseudo-labels as clean samples; the metric-based method first maps all samples to the feature space in an unsupervised manner, then determines whether a sample is clean or not by judging the pseudo-labels of the nearest k samples; and the hybrid-based method comprehensively considers the preceding two methods to divide clean and noise samples.

Arazo et al. [31] carried out an innovative work. Through previous works, it has been known that the loss of noise samples is greater than that of clean ones. Thus, they used a beta mixture model to fit the losses of all samples, finding that noise and clean images are separated in two areas with different mean values, and the clean and noise samples are divided.

2.2.3 Methods Based on Elaborate Model Structures. Methods based on elaborate model structures aim to design such structures to select clean samples for model training. Jiang et al. [33] proposed MentorNet, including a Mentor sub-network and a Student sub-network. The Mentor was pretrained for selecting clean instances to guide the updates of the Student. Since the training of a single network will produce biases, the error of noise labels will accumulate. Malach and Shalev-Shwartz [34] proposed the Decoupling model, which uses two sub-networks to select clean samples at the same time. To maintain the difference between the two sub-networks, only the images with inconsistent network predictions are used for training. Decoupling has two sub-networks, but they carry our forward propagation in one direction, which leads to the accumulation of errors. Han et al. [35] further proposed the Co-Teaching framework with two sub-networks. For each batch during training, first, two sub-networks communicate with each other what data in this batch should be used for training. Then, each sub-network backpropagates the data selected

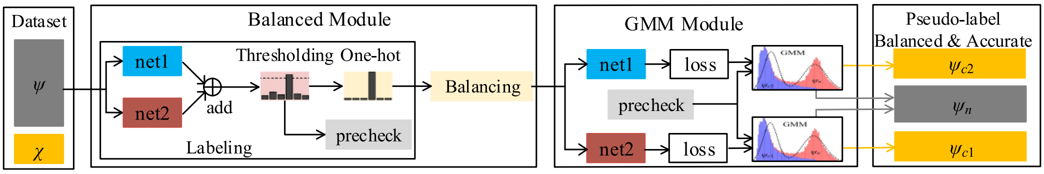


Fig. 1. Illustration of the proposed method. ⊕ refers to adding the outputs of net1 and net2; ψ is unlabeled data, whereas χ is labeled data; ψ_{c1} and ψ_{c2} refer to the clean examples from ψ divided by net1 and net2, respectively; and ψ_n refers to the noise examples from ψ divided by net1 and net2. First, the Labeling and Balancing operations in BM are performed on unlabeled data to obtain balanced pseudo-labels (Section 3.1). Then, in GMM, the losses of balanced dataset output by BM are fitted, and further considering the precheck information stored in BM, the dataset are divided into clean and noise parts (Section 3.2). Finally, the balanced and accurate training datasets are obtained for model training (Section 3.3).

by its peer network and updates itself. The cross-selecting mode in Co-Teaching makes the two sub-networks mutually exclusive, filters out noise labels, and alleviates the confirmation biases of deep networks.

3 THE PROPOSED METHOD

Traditional Co-Training and Co-Teaching both have two sub-networks, but Co-Training is used for semi-supervised learning, whereas Co-Teaching is used in LNL. The sub-networks in Co-Training will make pseudo-labels for each other, whereas the sub-networks in Co-Teaching divide the samples into clean and noise parts for other sub-networks. We design BAPS based on the structure of Co-Training and borrow ideas from Co-Teaching. Unlike traditional Co-Training, which only predicts the remaining unlabeled images in every K epochs of training, all unlabeled data will be labeled in BAPS every K epochs, and by drawing lessons from Co-Teaching, the high-confidence pseudo-labels are cross selected as clean ones from all unlabeled data for sub-network training.

The overall framework of BAPS is shown in Figure 1. To overcome the problem of an imbalanced training dataset and incorrect pseudo-labels, we designed BM and GMM. Let $D = \chi \cup \psi$ represent all images in the dataset, and $\chi = \{(x_i, y_i) : y_i \in \{0, 1\}^L\}_{i=1}^{N_\chi}$ are labeled samples, x_i and y_i represent the ith image and its one-hot label, L refers to the number of categories in D, and N_χ is the number of samples in D. $\psi = \{\varphi_i\}_{i=1}^{N_\psi}$ refers to unlabeled data, and N_ψ is the number of unlabeled images. First, use BM to predict and balance all unlabeled images to obtain a dataset with balanced pseudo-labels (Section 3.1). Then use GMM to divide the dataset output by BM into clean and noise collections (Section 3.2). Last, the outputs of GMM are used for network training (Section 3.3). During the training process, BM is executed every K epochs, whereas GMM will be carried out every epoch.

3.1 BM for Balanced Pseudo-Labels

The imbalance mentioned in this article is different from the one in traditional data mining [36]. The imbalance of the training dataset in this article is caused by the paranoia of the neural network itself, whereas the latter lies in the imbalanced characteristic of the dataset itself [37]. The reason pseudo-labels are not balanced is that the neural network will gradually become paranoid during the training process. Specifically, the network tends to classify those easy-to-classify images more quickly and correctly [30], such as the airplane, frog, and horse in CIFAR-10, but it is hard to classify the cat and dog, automobile, and car correctly. Thus, the number of labels for easy-to-classify images in pseudo-labels is greater than that for hard-to-classify images, which causes the imbalance distribution of the pseudo-labels. This imbalance is particularly serious at the early training stage and will greatly affect model training efficiency.

Through BM, balanced samples could be selected for model training. In every K epochs, all unlabeled data ψ will be forward propagated through the two sub-networks, and the two outputs are added together before converting into output probabilities through the softmax function. For image ϕ_i, the output probability q_i is defined as

$$q_i = soft\max\left(\sum_{k=1}^{2} P_{netk}\left(\phi\left(\varphi_i\right);\theta_k\right)\right), \tag{1}$$

where θ_k ($k \in \{1, 2\}$) represents the model parameters (weights and biases) of the sub-networks in BAPS; $P_{netk}(\varphi_i; \theta_k)$ is the output of ϕ_i through sub-network k; and $\phi(\cdot)$ and $\psi(\cdot)$ refer to weak and strong augmentation, respectively.

Generally, the Co-Training model adopts the voting method [17] to determine the guessed labels by the consistent predictions of the two sub-networks, whereas Equation (1) adds the two outputs and makes a final decision. As from the perspective of ensemble learning, by adding the output of multiple weak classifiers, we could get a better classification accuracy than any one of the classifiers. The output of softmax represents the possibility that sample ϕ_i is identified as a certain class in L. We set a confidence threshold τ of the clean samples, and if the maximum value of q_i (i.e., q_i.max) is greater than τ, then it can be inferred that the pseudo-label of ϕ_i is very likely to be $\hat{q}_i$. Based on this, we set the precheck information q_{prei} of ϕ_i to be 1 if q_i.max is greater than τ and otherwise 0. This q_{pre} will be used for the subsequent double verification in GMM. The pseudo-label of all samples satisfying the following condition could be computed by the following equation:

$$\hat{q}_i = \arg\max(q_i) \quad \text{if} \quad q_{prei} = 1. \tag{2}$$

The calculation process of BM is shown in Algorithm 1. We call the operation in Equations (1) and (2) *Labeling*, as described in lines 2 and 3, and lines 4 and 5 are the *Balancing* operation. After the Labeling operation, we can get the pseudo-labels of all unlabeled images, but there are many imbalanced noise labels.

To select a balanced number of samples, we first classify and count the pseudo-labels under the condition that $q_{prei} == 1$, and the least number of samples N_{min} among L classes is obtained. Then, sort the maximum output probabilities q.max of each category from high to low, and select the first rN_{min} samples as high-confidence samples to retain their pseudo-labels. The parameter r here determines the proportion of high-confidence images selected, and the setting of r is quite meaningful. To maintain the balance, we chose images based on the minimum value N_{min}. Considering that there are incorrect pseudo-labels in the N_{min} labels, we do not choose all N_{min} samples but only rN_{min} of them, whose output probability are the top rN_{min} largest one. r is set according to the training epochs, defined as follows:

$$r = \begin{cases} epoch\frac{0.4}{Epoch-20} + \frac{0.6Epoch-20}{Epoch-20} & epoch <= Epoch \\ 1 & epoch > Epoch \end{cases} \tag{3}$$

where *epoch* is the current number of iterations, and *Epoch* is a hyperparameter, which is less than the maximum number of iterations *MaxEpoch* of the model training process. From Equation (3), when *epoch* is less than *Epoch, r* gradually increases from 0.6 to 1 when *epoch* equals *Epoch.*

In this way, among all N_ψ samples, there are LrN_{min} ones with high-confidence pseudo-labels, and the number of samples in each category is rN_{min}. For the remaining N_ψ-LrN_{min} samples, a random label l, $l \in (1, \ldots, L)$ will be assigned. Thus, we could get a balanced pseudo-labeled dataset after BM. The calculation process of BM is shown in Algorithm 1. Classify and count the pseudo-labels under the condition that $q_{prei} == 1$, and the least number of samples N_{min} among L classes is obtained.

ALGORITHM 1: Computing process of BM

```
1: Input: $\theta_1$ and $\theta_2$, unlabeled data $\psi = \{\varphi_i\}_{i=1}^{N_\psi}$, clean probability threshold $\tau$, sample selection ratio $r$.
2: Calculate the output probability $q$ of all unlabeled data $\psi$ according to Equation (1), and determine whether $q_i$. max is greater than $\tau$ to get the precheck information $q_{prei}$. If $q_i$.max > $\tau$, then $q_{prei} = 1$, and otherwise $q_{prei} = 0$.
3: According to Equation (2), calculate pseudo-label $\hat{q}$ of all samples under the condition $q_{prei} == 1$.
4: Classify and count the pseudo-labels $\hat{q}$ to get the least number of samples $N_{min}$ among $L$ classes.
5: Sort the maximum output probability $q$.max of each category from high to low, and select the first $rN_{min}$ samples as high-confidence samples to retain their pseudo-labels, then label the remaining samples in $\psi$ as a random label $l$, $l \in (1, \ldots, L)$ to generate balanced pseudo-labels $\hat{q}_b$.
6: Return $\hat{q}_b$ and $q_{pre}$.
```

3.2 GMM for Accurate Pseudo-Labels

The training of randomly labeled images takes a longer time than that with correctly labeled ones [35, 38, 39], which means the losses of randomly labeled images will be greater than that of clean labels. Arazo et al. [31] and Li et al. [32] both use a mixed model to fit the losses of samples containing noise labels, and divide clean and noise samples by the fitting results. We also adopt a similar method to achieve this goal. Cross entropy could measure the fit of the network to unlabeled data $\psi = \{(\varphi_i : \hat{q}_{bi}) : \hat{q}_{bi} \in \{0,1\}^L\}_{i=1}^{N_\psi}$, with the balanced pseudo-labels obtained in BM. The cross entropy of image ϕ_i is calculated as

$$\ell_{CEi}(\theta) = -\hat{q}_{bi} \log\left(soft\max\left(P_{net}\left(\phi\left(\varphi_i\right);\theta\right)\right)\right) \tag{4}$$

The cross entropy of sample ϕ_i is simplified as ℓ_{CEi}, and for all of the images in ψ, we have $\ell_{CE} = \{\ell_{CEi}\}_{i=1}^{N_\psi}$. Then, we perform the normalization operation on ℓ_{CE} as follows:

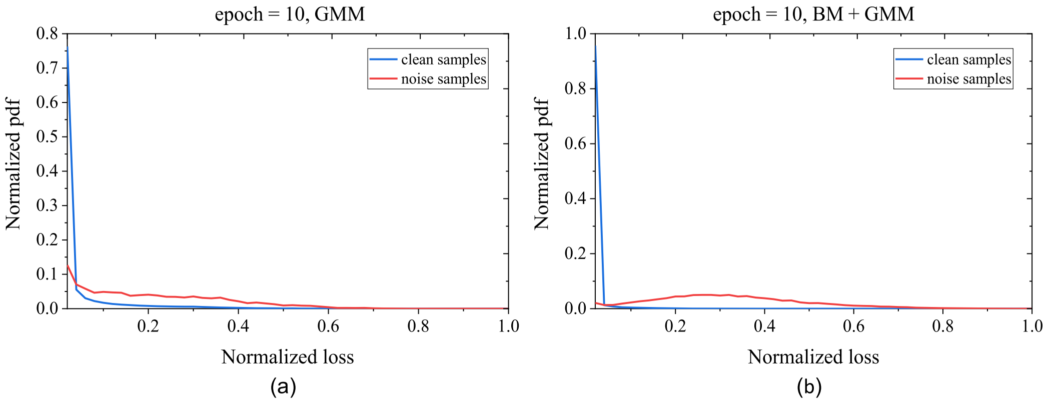
$$\ell_{CE} = \frac{\ell_{CE} - \ell_{CE}.\min}{\ell_{CE}.\max - \ell_{CE}.\min} \tag{5}$$

where $\ell_{CE}.\min$ and $\ell_{CE}.\max$ respectively are the minimum and maximum values of the cross-entropy loss in the N_ψ images. Since there are two types of clean and noise samples, we use a Gaussian mixture module consisting of two Gaussian functions to fit their losses and set the probability that the sample ϕ_i is clean to ω_i. As the losses of clean samples are small, we have

$$\omega_i = p(g/\ell_{CEi}) \tag{6}$$

where g is the Gaussian function whose output probability is the smaller one. Since there are only two Gaussian functions, we set a threshold of 0.5 for ω_i. Taking both ω_i and q_{pre} into consideration, we classify ϕ_i as clean, and the pseudo-label $\hat{q}_{bi}$ of it to $\hat{q}_{bci}$, and otherwise as a noise sample.

It should be noted that in the LNL task of Li et al. [32], the authors not only consider the cross-entropy loss but also the entropy of the images due to the asymmetric noises (imbalanced characteristic of incorrectly labeled images). In this article, we have elaborately designed BM, so the influence of asymmetric noises has been overcome greatly, and we only use cross-entropy loss to the fitting process. The following experiments test the influence of BM on the fitting effect of GMM. In Figure 2(a), GMM fits the losses by net1 with all images whose pseudo-labels are obtained in the Labeling operation in BM, whereas in Figure 2(b), the pseudo-labels are the outputs of BM. Comparing Figure 2(a) and (b), it can be seen that it is difficult for GMM to fit the cross-entropy loss of samples without the operation of Balancing due to the asymmetric errors in pseudo-labels, but for the balanced output of BM, there are two distinct peaks to distinguish the clean and noise samples.



(a) (b)

Fig. 2. After training for 10 epochs, the fitting effect of GMM on all pseudo-labeled images: without the Balancing operation in BM (a) and with BM (b).

3.3 Training with Balanced and Accurate Samples

BM and GMM are the two core modules of the Deep Co-Training model proposed in this article. Through BM and GMM, a clean dataset with balanced and accurate pseudo-labeled images are obtained, and we further designed the loss for model training. For labeled samples χ, we directly compute the cross-entropy loss as follows:

$$\ell_{\chi}(\theta) = -\frac{1}{N_{\chi}} \sum_{i=1}^{N_{\chi}} y_i \log(soft\max(P_{net}(\phi(x_i);\theta))) \tag{7}$$

For clean samples $\psi_c = \{(\varphi_{ci} : \hat{q}_{bci}) : \hat{q}_{bci} \in \{0,1\}^L\}_{i=1}^{N_c}$ divided by GMM, N_c represents the number of clean samples. We did not use $\hat{q}_{bc}$ directly, but first used sub-networks 1 and 2 to predict them again. If the guessed labels are consistent with $\hat{q}_{bc}$, then they will be used as a supervisory signal for cross-entropy loss calculation. This idea is the same as that of Nguyen et al. [30], which means the predicted volatility of clean samples is more stable than noise ones. Thus, for ψ_c, the loss is defined as follows:

$$\ell_{\psi_c}(\theta) = -\frac{1}{N_c} \sum_{i=1}^{N_c} \mathrm{I}(\hat{q}_{bi} == \hat{q}_{bci})\hat{q}_{bci} \log(soft\max(P_{net1}(\psi(x_i);\theta))) \tag{8}$$

where I(·) is an indicator function. If the condition is true, the returning value is 1, and otherwise 0. During the training process of sub-network 1, the forward propagation output of net1 $P_{net1}(\phi(x_i);\theta)$ needs to carry out error backpropagation for model parameter updating of sub-network 1, but there is no need for $P_{net2}(\phi(x_i);\theta)$. Equation (8) is used for the training of sub-network 1, and it is the same situation for the training of sub-network 2.

For noise samples $\psi_n = \{\varphi_{ni}\}_{i=1}^{N_n}$, since most of the N_n samples are difficult to obtain correct pseudo-labels with high-confidence, they will bring in wrong supervisory information if used in the same way as Equation (8). Thus, we introduce KL divergence [21] to minimize the predictions between $\phi(\varphi_{ni})$ and $\psi(\varphi_{ni})$, and as sub-networks 1 and 2 can be combined to get better prediction results, we get the following loss:

$$\ell_{\psi_n}(\theta) = D_{KL}\left(\frac{1}{2}\sum_{k=1}^{2} P_{netk}(\phi(\varphi_{ni});\theta_k)||P_{net1}\,(\psi(\varphi_{ni});\theta_1)\right) \tag{9}$$

where $D_{KL}(\cdot)$ refers to KL divergence. During the training process of sub-network 1, the forward propagation outputs of net2 $P_{net2}(\phi(x_i);\theta)$ do not need to carry out error backpropagation. Equation (9) is used for the training of sub-network 1, and it is the same situation for the training of sub-network 2.

Finally, the total loss ℓ is

$$\ell = \ell_{\chi} + \lambda_{u1}\ell_{\psi_c} + \lambda_{u2}\ell_{\psi_n} \tag{10}$$

where λ_{u1} and λ_{u2} are the weights to control the strength of ℓ_{ψ_c} and ℓ_{ψ_n}. As the noise samples divided by GMM accounted for a large proportion in the initial training stage, we directly calculate the KL loss in our subsequent experiments on the whole dataset corresponding to Equation (9).

Due to the poor division effect of sub-networks at the initial stage of training, a training scheduler similar to that of Li et al. [32] is adopted to pretrain the two sub-networks for t epochs. Equations (7) through (9) are still used in the pretraining stage, but we do not provide pseudo-labels for Equation (8), and we adopt the way used in FixMatch for this t epochs. In addition, drawing lessons from ensemble learning, we add the predictions of the two sub-networks for the testing process, and the recognition result is the category corresponding to the maximum value of the output.

Algorithm 2 delineates the full algorithm of BAPS.

ALGORITHM 2: Balanced and accurate pseudo-labels for semi-supervised image classification

```
Input: $\theta_1$ and $\theta_2$, training dataset D = $\chi \cup \psi$, clean probability threshold $\tau$, number of pretrained epochs $t$, maximum epochs $MaxEpoch$, BM execution frequency $K$, sample selection ratio r, unsupervised loss weight $\lambda_{u1}, \lambda_{u2}$
$\theta_1, \theta_2$ = Pretrain($D$, $\theta_1$, $\theta_2$).
while $epoch < MaxEpoch$ do
    if $epoch < t$ then
        $\theta_1, \theta_2$ = Pretrain($D$, $\theta_1$, $\theta_2$).
    else
        if $epoch \% K == 0$ then
            $\hat{q}_b, q_{pre}$ = BM($\psi$, $\theta_1$, $\theta_2$, $\tau$, r).
        end if
        $\omega_2$ = GMM($\psi, \theta_1, \hat{q}_b, q_{pre}$).
        $\omega_1$ = GMM($\psi, \theta_2, \hat{q}_b, q_{pre}$).
        for $k = 1, 2$ do
            $\psi_{ck} = \{(\varphi_{ci}, \hat{q}_{bci}) | \omega_{ki} > 0.5, \forall(\varphi_{ci}, \hat{q}_{bci}, \omega_{ki}) \in (\psi, \hat{q}_b, \omega_k)\}_{i=1}^{N_{ck}}$.
            $\psi_{nk} = \{\varphi_{ni} | \omega_{ki} \le 0.5, \forall(\varphi_{ni}, \omega_{ki}) \in (\psi, \omega_k)\}_{i=1}^{N_{nk}}$.
            Calculate cross-entropy loss of $\chi$ and $\psi_{ck}$ according to Equations (7) and (8), and KL loss of $\psi_{nk}$
             according to Equation (9).
            $\ell = \ell_{\chi} + \lambda_{u1}\ell_{\psi_c} + \lambda_{u2}\ell_{\psi_n}$.
            $\theta_k = SGD(\ell, \theta_k)$.
        end for
    end if
end while
```

4 EXPERIMENTS

This section first introduces the dataset and model parameter settings used in BAPS, then gives the experimental results of the proposed algorithm and other state-of-the-art semi-supervised methods on CIFAR-10/100 and SVHN, and finally, the ablation studies of the important components and hyperparameters are carried out.

4.1 Dataset and Experiment Setups

Datasets. We extensively validate our method on three benchmark datasets, namely CIFAR-10/100 [40] and SVHN [41]. Both CIFAR-10 and CIFAR-100 contain 50K training images and 10K test images of size 32 × 32. CIFAR-10 contains 10 classes of images, and we randomly select 400 images from each class in the training images to construct a training dataset containing 4K labeled images and 46K unlabeled ones. CIFAR-100 contains 100 classes of images, and we randomly select 100 images from each class in the training images to form a training dataset containing 10K labeled images and 40K unlabeled images. SVHN contains 10 categories of images; the number of training and testing images is 73,257 and 26,032, respectively; and we randomly select 100 images from each category in the training dataset as the labeled ones.

Parameters. For CIFAR-10 and SVHN, we chose Wide-ResNet-28-2 [42, 43] with 1.47M parameters as the deep structure of sub-networks. As CIFAR-100 contains more classes and requires a larger network capacity, we use Wide-ResNet-28-8 including 23.4M parameters. We implement the proposed algorithm using the PyTorch framework. All experiments are conducted on a server consisting of two 2080Ti GPUs with 11 GB of memory each.

We train our Deep Co-Training model using SGD with a Nesterov momentum of 0.9 and a weight decay with a fixed value of 5×10^{-4}. Similar to Xie et al. [21], we set a batch size of 64 to labeled images, and 64μ to unlabeled ones, and set μ to 7. The network is trained for *MaxEpoch* epochs with 1,024 steps each, and the pretrained epoch t is 20 for CIFAR-10 and SVHN and 30 for CIFAR-100. We set the initial learning rate as 0.03 and use the same learning rate schedule as FixMatch. For all experiments, we use the same hyperparameters of $K = 10$, $\tau = 0.9$, $Epoch = 500$, $MaxEpoch = 600$, $\lambda_{u1} = 1$, and $\lambda_{u2} = 1$.

For every epoch, we use an exponential moving average over the two sub-networks weights with a decay of 0.999, and the forward propagation process in BM and GMM are calculated using the EMA model; the test results are also obtained by the EMA model. The weak augmentation in this article is a flip operation that randomly flips the input images horizontally with a probability of 50%. For strong augmentation policy, we chose RandAugment [44], which has been used in state-of-the-art semi-supervised algorithms. RandAugment includes a series of transformations, including color inversion, translation, and contrast adjustment, among others. During the training process, RandAugment randomly chooses two transforms from the collections for each mini-batch.

4.2 Comparison of Experimental Results

We use classification accuracy as a measure of various algorithms. The state-of-the-art semi-supervised algorithms are selected for comparison, including Co-Training-based algorithms DCT [16], tri-net [17], and PLW-ML [18]; Consistency Regularization based algorithms, including π-Model [19], Mean Teacher [20], UDA [21], FixMatch [22], and ACL [49]; algorithms based on Mixup, such as MixMatch [24], ReMixMatch [25], SelfMatch [45], and RML-CNN [48]; and methods based on adversarial training, such as VAT [46] and CCS-GAN [47]. The classification accuracy is shown in Table 1. In both 4K-label experiments on CIFAR-10 and 10K-label experiments on CIFAR-100, the proposed BAPS outperforms all state-of-the-art semi-supervised image classification methods. But in the 1K-label experiment on the imbalanced dataset SVHN, the accuracy of BAPS is slightly worse than that of FixMatch, UDA, SelfMatch, and ReMixMatch.

4.3 Ablation Study for BAPS

BAPS is a Deep Co-Training model that consists of two sub-networks. To illustrate the function of each component, we removed BM, GMM, and KL loss, respectively, and conducted comparative experiments on the CIFAR-10 dataset. In addition, the parameter τ is quite important for the

Table 1. Comparison of Classification Accuracy on Different Datasets (%)

Method	SVHN	CIFAR-10	CIFAR-100
π-Model (2017)	92.46 ± 0.36	85.99 ± 0.38	62.12 ± 0.11
Mean Teacher (2017)	96.58 ± 0.11	90.81 ± 0.19	64.17 ± 0.24
DCT(2 views) (2018)	96.39 ± 0.15	90.97 ± 0.18	61.23 ± 0.28
Tri-net (2018)	96.29 ± 0.14	91.55 ± 0.22	–
MixMatch (2019)	96.5 ± 0.28	93.58 ± 0.10	71.69 ± 0.33
VAT (2019)	94.58 ± 0.22	88.64 ± 0.34	–
UDA (2020)	97.54 ± 0.24	95.12 ± 0.18	75.5 ± 0.25
FixMatch(RA) (2020)	**97.72** ± 0.11	95.74 ± 0.05	77.4 ± 0.12
ReMixMatch (2020)	97.36 ± 0.08	95.28 ± 0.13	76.97 ± 0.56
RML-CNN (2020)	–	91.46 ± 0.20	–
SelfMatch (2021)	97.49 ± 0.07	95.94 ± 0.08	–
PLW-ML (2021)	95.45 ± 0.15	93.12 ± 0.23	–
CCS-GAN (2021)	95.64 ± 0.09	85.99 ± 0.15	67.85 ± 0.44
ACL (2021)	94.83 ± 0.21	89.78 ± 0.72	–
BAPS	97.27 ± 0.19	**96.35** ± 0.05	**79.45** ± 0.08

selection of pseudo-labels, and we have also discussed this parameter. We also conduct comparative experiments with BAPS models with 3, 4, and 5 subnetworks.

On BM. In BAPS, BM could overcome the confirmation bias of deep neural networks to a certain extent and obtain the balanced pseudo-labels. In this experiment, to verify the function of BM on the whole BAPS model, we removed the Balancing operation in BM; specifically, the samples with high-confidence pseudo-labels are selected according to sample selection ratio r, then passed to GMM. The control group experiment keeps the BAPS unchanged and uses the default parameter settings ($K = 10$, $\tau = 0.9$, $Epoch = 500$, $MaxEpoch = 800$, $\lambda_{u1} = 1$, and $\lambda_{u2} = 1$).

Figure 3(a) shows the change of classification accuracy of the test set during the training process with and without BM. It can be seen that the original model could get a better classification accuracy than the test model without BM. We further show the distributions of the samples with correct pseudo-labels (the result of comparisons with the true labels of the samples, this ground truth is only used to obtain the visualization results in the following figures, not for model training) that provide correct supervisory information in the training dataset ψ_{c1} when the epoch is 100, 300, and 500, respectively. From Figure 3(b), we can see that the sample distribution in the original model with BM is almost balanced throughout the training process; however, for the test model without BM, the samples that provide a correct supervisory signal are not balanced, although this imbalance will gradually be relieved as the training progresses, but it will still affect the training process and reduce training efficiency.

On GMM. GMM could be used to fit the losses of samples with pseudo-labels that are correctly and incorrectly labeled, so as to achieve the purpose of dividing clean and noise samples. BAPS is designed to cross-divide samples, and to maintain this Co-Teaching mechanism when GMM is removed, we make sub-networks cross-divide samples for each other directly, then use the Balancing operation in BM to obtain balanced samples for model training. The control group experiment keeps the model unchanged, and the parameters use the default parameter settings ($K = 10$, $\tau = 0.9$, $Epoch = 500$, $MaxEpoch = 600$, $\lambda_{u1} = 1$, and $\lambda_{u2} = 1$).

Figure 4(a) shows the change of classification accuracy of the test set during the training process before and after removing GMM, and we can easily draw conclusions that the classification accuracy of the model without GMM is worse than that of BAPS. We also show the proportion

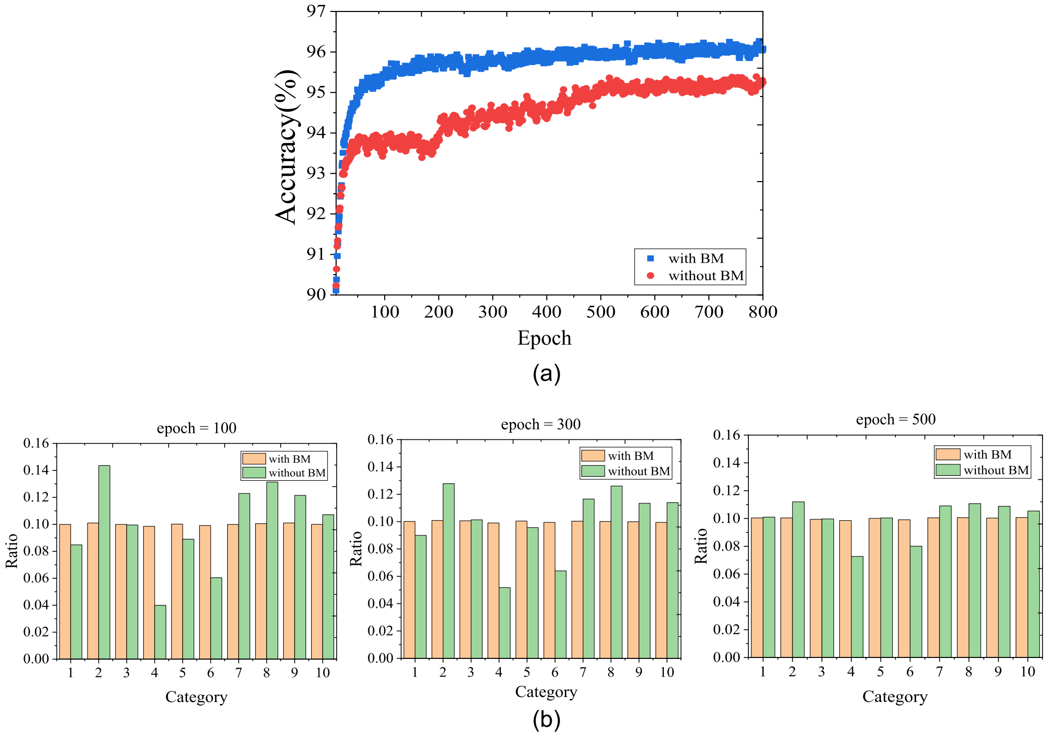


Fig. 3. (a) Comparison of the classification accuracy of the model in the test set with or without BM. (b) The distributions of the samples that provide correct supervised information in training dataset ψ_{c1} when the epoch is 100, 300, and 500, respectively.

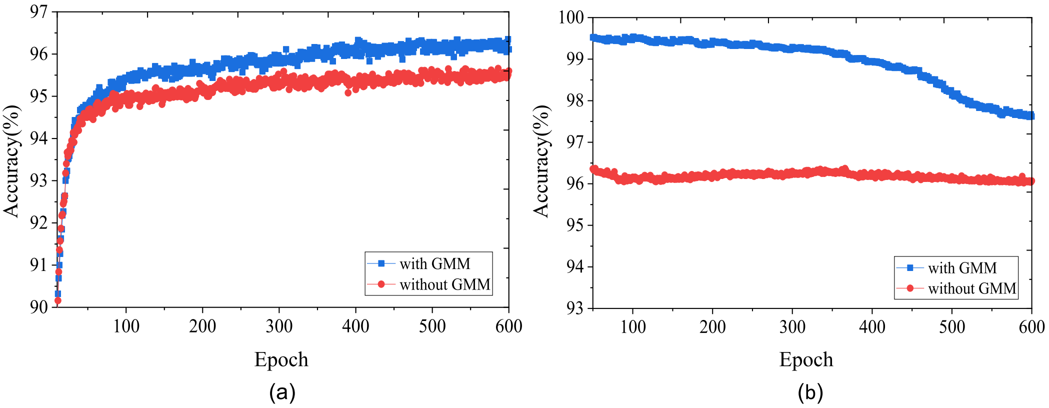


Fig. 4. (a) Comparison of the classification accuracy of the model in the test set with or without GMM. (b) The proportion of samples that provide correct supervisory information in training dataset ψ_{c1}.

of samples with correct pseudo-labels that provide an accurate supervisory signal in ψ_{c1}. From Figure 4(b), the proportion of samples that provide correct supervisory information in ψ_{c1} exceeds 99% for BAPS and only 96% for the test model without GMM. In addition, as training progresses, that ratio in BAPS slightly reduces due to the gradually increases of sample selection ratio r, and

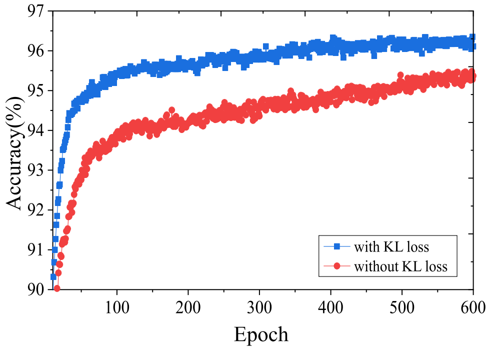


Fig. 5. Comparison of the classification accuracy of the model in the test set with or without KL loss.

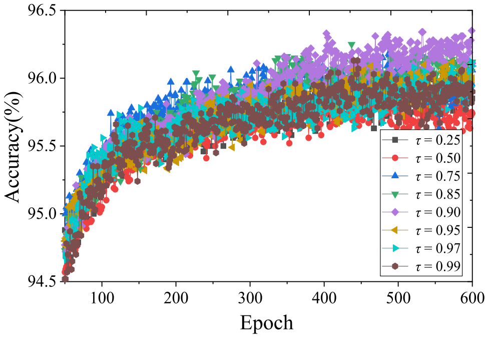


Fig. 6. Comparison of the classification accuracy of the model in the test set when τ is chosen from {0.25, 0.50, 0.75, 0.85, 0.90, 0.95, 0.97, 0.99}.

when *epoch* reaches 500, the value of r will be 1.0, which means the number of samples in ψ_{c1} is getting closer and closer to N_ψ, and more and more samples are difficult to classify as clean ones with high confidence. Even if the division results become worse in the later training stage, the proportion of clean samples in BAPS is much higher than that in the test model without GMM.

On KL loss. As the cross-entropy loss ℓ_{ψ_c} with the clean samples may bring in interferences, we introduced ℓ_{ψ_n}. Set $\lambda_{u2} = 0$ for the test group and $\lambda_{u2} = 1$ for the control group, and use the default parameters for the rest of the parameters ($K = 10$, $\tau = 0.9$, $Epoch = 500$, $MaxEpoch = 600$, and $\lambda_{u1} = 1$). It can be seen from Figure 5 that the classification accuracy of the BAPS model with KL loss on the test set is higher than that of the test model without KL loss, indicating that KL loss promotes the training of the BAPS model.

On the confidence threshold τ. The value of τ is crucial for choosing pseudo-labels. We set the values of τ to 0.25, 0.5, 0.75, 0.85, 0.90, 0.95, 0.97, and 0.99, respectively, for comparison experiments, and the remaining parameters adopt the default parameter settings ($K = 10$, $Epoch = 500$, $MaxEpoch = 600$, $\lambda_{u1} = 1$, and $\lambda_{u2} = 1$). It can be seen from Figure 6 that as the value of τ increases from 0.25 to 0.99, the accuracy of the test set shows a trend of first increasing and then decreasing, and when τ is set to 0.9, the BAPS model can obtain the best result on the test set. Thus, we set τ to 0.9.

On the ensemble learning mechanism. There are two sub-networks in our BAPS for co-training. Considering the idea of ensemble learning, by adding the output of multiple weak classifiers, we could get better classification accuracy than any one of its sub-classifiers. We conduct comparative experiments on CIFAR10, and the number of sub-networks in BAPS is set to 2, 3, 4, and 5, respectively. In BM, the unlabeled samples are pseudo labeled by averaging the outputs of all sub-networks by formula (1), and the division of the training samples of a sub-network in GMM is obtained by all other sub-networks. The other parameter settings are $K = 10$, $Epoch = 500$, $MaxEpoch = 100$, $\lambda_{u1} = 1$, and $\lambda_{u2} = 1$.

The change of BAPS accuracy with different numbers of sub-networks is shown in Figure 7(a), and Figure 7(b) through (e) show the changes of the BAPS accuracy with its sub-networks when the number of sub-networks is 2, 3, 4, and 5, respectively. It can be seen from Figure 7(a) that the more the numbers of sub-networks, the higher the accuracy of BAPS, but when the epoch is <70, the accuracy of BAPS with two sub-networks is a little higher than that with three sub-networks, which may be caused by the instability in the early training stage of the model. When the epoch is >70, the accuracy of the model with three sub-networks exceeds that of the model with two sub-networks. From the comparison of Figure 7(b) to Figure 7(e), the accuracy of BAPS is higher

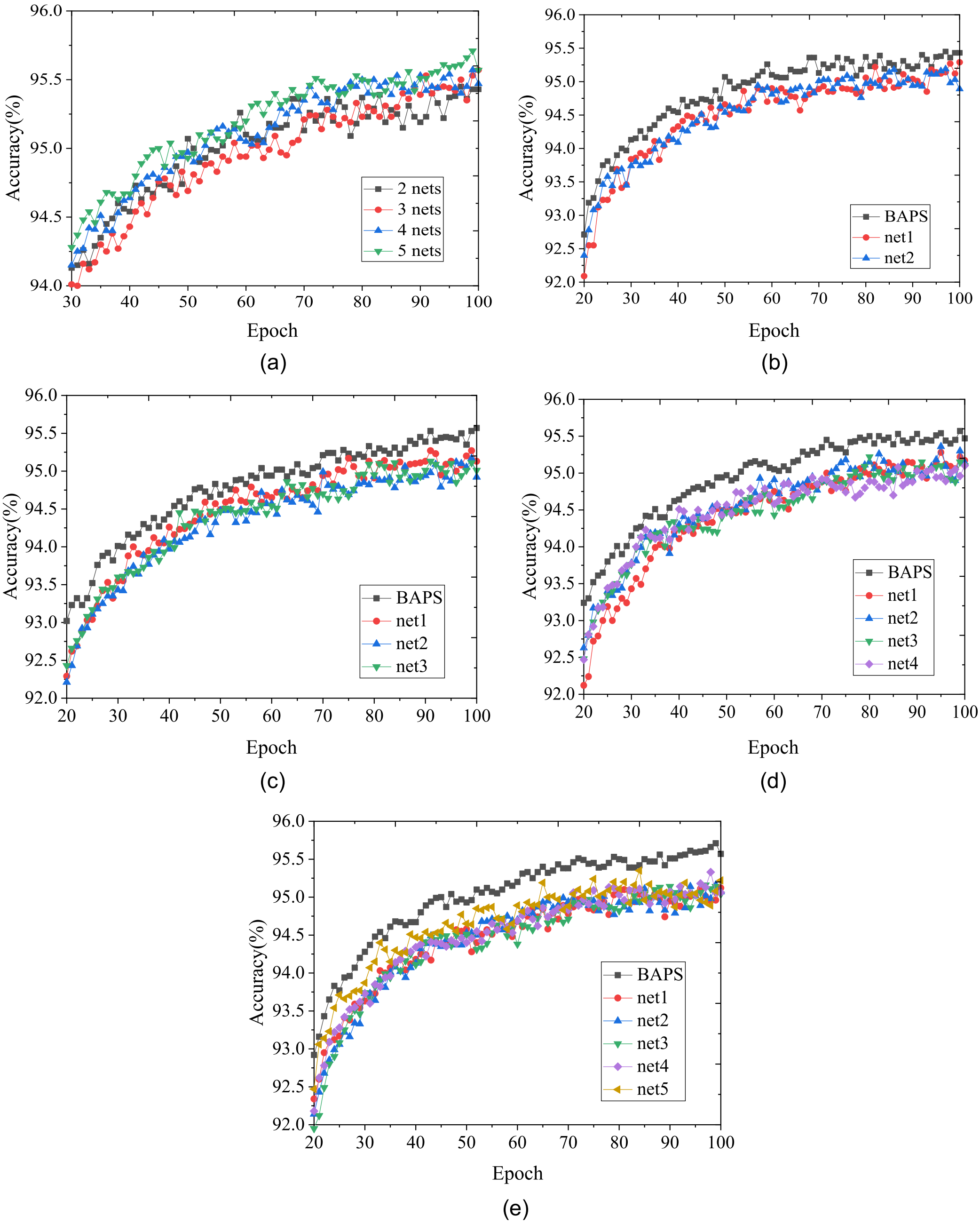


Fig. 7. (a) Comparison of the classification accuracy of BAPS in the test set when the number of sub-networks is 2, 3, 4, and 5, respectively. (b) Comparison of the classification accuracy of BAPS with its two sub-networks in the test set. (c) Comparison of the classification accuracy of BAPS with its three sub-networks in the test set. (d) Comparison of the classification accuracy of BAPS with its four sub-networks in the test set. (e) Comparison of the classification accuracy of BAPS with its five sub-networks in the test set.

than any one of its sub-networks, and this trend becomes larger as the number of sub-networks increases. In addition, the accuracy between the sub-networks is basically the same.

5 CONCLUSION

Co-Training is an important framework for semi-supervised image classification. The sub-networks in this structure will first label the unlabeled images and then select the images with high-confidence pseudo-labels to put them into each other's training set. But this traditional cross-labeling operation will bring in imbalanced noise labels. This article proposed BAPS with BM and GMM components. In BM, two sub-networks are first used to jointly predict unlabeled samples and then samples with high confidence are selected for the Balancing operation to obtain balanced samples of each category. Referring to the common practice of the LNL task, we use GMM to fit the loss distribution of the samples output by BM, then clean samples and noise samples are divided based on the fact that the loss of correctly labeled samples is generally smaller than that of wrongly labeled ones. At the same time, referring to the training mechanism of Co-Teaching, we design the two sub-networks in GMM to divide samples for each other during every iteration to avoid accumulation of errors. Through a large number of comparative experiments, the Deep Co-Training model of this article has achieved very good classification accuracy on CIFAR-10/100 and SVHN datasets, and further ablation experiments demonstrate the effectiveness of BAPS.

However, our BAPS focuses on alleviating the imbalance problem from the deep model itself and cannot get the best classification performance in class-imbalanced dataset such as SVHN, which is a shortcoming of our BAPS. Next, we will focus on improving the performance of our model on long-tail datasets.

REFERENCES

[1] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. MobileNetV2: Inverted residuals and linear bottlenecks. *arXiv preprint arXiv:1801.04381* (2018).

[2] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. 2019. Squeeze-and-excitation networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 8 (2019), 2011–2023.

[3] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. 2017. Learning transferable architectures for scalable image recognition. *arXiv preprint arXiv:1707.07012* (2017).

[4] Yunpeng Chen, Jianan Li, Huaxin Xiao, Xiaojie Jin, Shuicheng Yan, and Jiashi Feng. 2017. Dual path networks. *arXiv preprint arXiv:1707.01629* (2017).

[5] Xingcheng Zhang, Zhizhong Li, Chen Change Loy, and Dahua Lin. 2017. PolyNet: A pursuit of structural diversity in very deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*. 3900–3908.

[6] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934* (2020).

[7] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Quyang, and Dahua Lin. 2019. Libra R-CNN: Towards balanced learning for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'19)*. 821–830.

[8] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. 2017. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 12 (2017), 2481–2495.

[9] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. 2017. Fully convolutional instance-aware semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*. 2359–2367.

[10] Scott E. Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2015. Training deep neural networks on noisy labels with bootstrapping. In *Proceedings of the International Conference on Learning Representations (ICLR'15)*. 1–11.

[11] Hongyu Guo and Herna L. Viktor. 2004. Learning from imbalanced data sets with boosting and data generation: The DataBoost-IM approach. *ACM SIGKDD Explorations Newsletter* 6, 1 (2004), 30–39.

[12] Bartosz Krawczyk, Mikel Galar, Lukasz Jelen, and Francisco Herrera. 2016. Evolutionary undersampling boosting for imbalanced classification of breast cancer malignancy. *Applied Soft Computing* 38 (2016), 714–726.

[13] Yun Qian, Yanchun Liang, Mu Li, Guoxiang Feng, and Xiaohu Sha. 2014. A resampling ensemble algorithm for classification of imbalance problems. *Neurocomputing* 143 (2014), 57–67.

[14] Yun Hon, Li Li, Bailin Li, and Jiajia Liu. 2019. An anti-noise ensemble algorithm for imbalance classification. *Intelligent Data Analysis* 23, 6 (2019), 1205–1217.

[15] Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*. 92–100.

[16] Siyuan Qiao, Wei Shen, Zhishuai Zhang, Bo Wang, and Alan Yuille. 2018. Deep Co-Training for semi-supervised image recognition. In *Proceedings of the European Conference on Computer Vision (ECCV'18)*. 1–17.

[17] Dongdong Chen, Wei Wang, Wei Gao, and Zhihua Zhou. 2018. Tri-net for semi-supervised deep learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*. 2014–2020.

[18] Jianwen Mo, Yuwan Gan, and Hua Yuan. 2021. Weighted pseudo labeled data and mutual learning for semi-supervised classification. *IEEE Access* 9 (2021), 36522–36534.

[19] Samuli Laine and Timo Aila. 2017. Temporal ensembling for semi-supervised learning. In *Proceedings of the International Conference on Learning Representations (ICLR'17)*. 1–13.

[20] Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS'17)*. 1–16.

[21] Qizhe Xie, Zihang Dai, Eduard Hovy, Minhthang Luong, and Quoc V. Le. 2020. Unsupervised data augmentation for consistency training. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS'20)*. 1–19.

[22] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. 2020. FixMatch: Simplifying semi-supervised learning with consistency and confidence. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS'20)*. 1–21.

[23] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. Mixup: Beyond empirical risk minimization. In *Proceedings of the International Conference on Learning Representations (ICLR'18)*. 1–13.

[24] David Berthelot, Nicholas Carlini, Ian Goodfellow, Avital Oliver, Nicolas Papernot, and Colin Raffel. 2019. MixMatch: A holistic approach to semi-supervised learning. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS'19)*. 1–14.

[25] David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, Colin Raffel, and Kihyuk Sohn. 2020. ReMixMatch: Semi-supervised learning with distribution alignment and augmentation anchoring. In *Proceedings of the International Conference on Learning Representations (ICLR'20)*. 1–13.

[26] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems* 27, 2672-2680.

[27] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training GANs. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS'16)*. 1–11.

[28] Augustus Odena. 2016. Semi-supervised learning with generative adversarial networks. In *Proceedings of the International Conference on Machine Learning (ICML'16)*. 1–3.

[29] Sungwon Park, Sungwon Han, Sundong Kim, Danu Kim, Sungkyu Park, Seunghoon Hong, and Meeyoung Cha. 2021. Improving unsupervised image clustering with robust learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'21)*. 12278–12287.

[30] Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. 2020. SELF: Learning to filter noisy labels with self-ensembling. In *Proceedings of the International Conference on Learning Representations (ICLR'20)*. 1–15.

[31] Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinnes. 2019. Unsupervised label noise modeling and loss correction. In *Proceedings of the International Conference on Machine Learning (ICML'19)*. 1–12.

[32] Junnan Li, Richard Socher, and Steven C. H. Hoi. 2020. DIVIDEMIX: Learning with noisy labels as semi-supervised learning. In *Proceedings of the International Conference on Learning Representations (ICLR'20)*. 1–14.

[33] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Lijia Li, and Feifei Li. 2018. MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *Proceedings of the International Conference on Machine Learning (ICML'18)*. 1–10.

[34] Eran Malach and Shai Shalev-Shwartz. 2017. Decoupling "when to update" from "how to update." In *Proceedings of the Conference on Neural Information Processing Systems (NIPS'17)*. 1–19.

[35] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS'18)*. 1–11.

[36] Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. 2021. Improving calibration for long-tailed recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'21)*. 16489–16498.

[37] Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Yuille, and Fan Yang. 2021. CReST: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'21).* 10857–10866.

[38] Devansh Arpit, Stanislaw Jastrz ebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, et al. 2017. A closer look at memorization in deep networks. In *Proceedings of the International Conference on Machine Learning (ICML'17).* 233–242.

[39] Pengfei Chen, Benben Liao, Guangyong Chen, and Shengyu Zhang. 2019. Understanding and utilizing deep neural networks trained with noisy labels. In *Proceedings of the International Conference on Machine Learning (ICML'19).* 1062–1070.

[40] A. Krizhevsky. 2009. *Learning Multiple Layers of Features from Tiny Images. Technical Report.* University of Toronto.

[41] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. 2011. Reading digits in natural images with unsupervised feature learning. In *Proceedings of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning.*

[42] Sergey Zagoruyko and Nikos Komodakis. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146* (2016).

[43] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16).* 770–778.

[44] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le. 2019. RandAugment: Practical automated data augmentation with a reduced search space. *arXiv preprint arXiv:1909.13719* (2019).

[45] Byoungjip Kim, Jinho Choo, Y.-D. Kwon, S. Joe, S. Min, and Y. Gwon. 2021. SelfMatch: Combining contrastive self-supervision and consistency for semi-supervised learning. *arXiv preprint arXiv:2101.06480* (2021).

[46] Takeru Miyato, Shin-Ichi Maeda, Masanori Koyama, and Shin Ishii. 2019. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 8 (2019), 1979–1993.

[47] Lei Wang, Yu Sun, and Zheng Wang. 2021. CCS-GAN: A semi-supervised generative adversarial network for image classification. *Visual Computer.* Published online, 2021.

[48] Hakan Cevikalp, Burak Benligiray, and Omer Nezih Gerek. 2020. Semi-supervised robust deep neural networks for multi-label image classification. *Pattern Recognition* 100 (2020), 107164.

[49] Jiaming Chen, Meng Yang, and Jie Ling. 2021. Attention-based label consistency for semi-supervised deep learning based image classification. *Neurocomputing* 453 (2021), 731–741.

Received 27 September 2021; accepted 13 December 2021