International Journal of Data Warehousing and Mining
Volume 19 • Issue 2

Top-K Pseudo Labeling for Semi-Supervised Image Classification

Yi Jiang, Harbin University of Science and Technology, China*
https://orcid.org/0000-0003-2800-9114

Hui Sun, Harbin University of Science and Technology, China

ABSTRACT

In this paper, a top-k pseudo labeling method for semi-supervised self-learning is proposed. Pseudo labeling is a key technology in semi-supervised self-learning. Briefly, the quality of the pseudo label generated largely determined the convergence of the neural network and the accuracy obtained. In this paper, the authors use a method called top-k pseudo labeling to generate pseudo label during the training of semi-supervised neural network model. The proposed labeling method helps a lot in learning features from unlabeled data. The proposed method is easy to implement and only relies on the neural network prediction and hyper-parameter k. The experiment results show that the proposed method works well with semi-supervised learning on CIFAR-10 and CIFAR-100 datasets. Also, a variant of top-k labeling for supervised learning named top-k regulation is proposed. The experiment results show that various models can achieve higher accuracy on test set when trained with top-k regulation.

KEYWORDS

Deep Convolutional Network, Deep Learning, Image Classification, Neural Network, Pseudo Label, Self-Training, Semi-Supervised Learning, Soft Label, Supervised Learning

INTRODUCTION

Semi-supervised learning is a special form of machine learning and deep learning method. The aim of semi-supervised learning is to utilize the unlabeled data that are often easy and cheap to be obtained. In practice, the availability of labeled data and unlabeled is asymmetrical. The unlabeled data is easy to obtain and large in amount, but there are fewer ways to utilize them fully. Semi-supervised learning solves the problem by training the classifier using both the labeled data and unlabeled data at the same time. Semi-supervised learning has recently become more popular and practically relevant due to the variety of problems for which vast quantities of unlabeled data are available, such as text on Web sites, protein sequences, or images (Zhu, 2005). Semi-supervised learning can be categorized into generative model (Kingma et al., 2014; Pu et al., 2016), co-training method (Blum & Mitchell,

DOI: 10.4018/IJDWM.316150 *Corresponding Author

1998; Sindhwani & Rosenbery, 2008), and self-training method (Berthelot et al., 2019; Lee, 2013; Sohn et al., 2020; Xie et al., 2020; Zhang et al., 2018). In recent years, researchers have focused more on the self-training method.

Generally, the idea of self-learning is to first train the classifier with labeled data, and then to use the trained classifier to classify, and get a predicted label for unlabeled data. In this study, the predicated label of unlabeled data can be used as the so-called pseudo label. Finally, the labeled data with real label and the unlabeled data with pseudo label can be used to train the classifier in the next iteration until the classifier converges. The ideal of self-learning is also that the quality of the classifier is largely determined by the quality of the pseudo label generated, and vice versa.

The pseudo label can be classified into hard (pseudo) label and soft (pseudo) label. The soft label method is to use the neural network prediction as the label of unlabeled data in semi-supervised training, which means the soft label is the SoftMax output of the model and represents the likelihood that unlabeled data belong to each class. The hard label method is to choose the class with the highest probability in SoftMax prediction as the label of unlabeled data (the hard label is a one-hot vector in most of the cases). It is clear that both soft label and hard label are inaccurate for unlabeled data. The soft label contains too much noises, and the hard label may be the incorrect with high probability. Both the noisy soft label and the incorrect hard label will prevent the semi-supervised model from being convergent.

In this study, the authors focused on the generating of pseudo label for self-learning based semi-supervised learning, and proposed a new method to generate pseudo label. The new method is named top-k labeling (or top-k label guessing). Briefly, the proposed labeling method chooses the top-k elements in model prediction as pseudo label, and sets the other elements in pseudo label to zero. This method is different from existing methods. Also, the proposed method can collaborate with any existing semi-supervised learning algorithms that were designed to train neural model with labeled data and unlabeled data at the same time, such as Mixup (Zhang et al., 2018) and MixMatch (Berthelot et al., 2019).

Among all the self-learning based semi-supervised learning algorithms, the authors thought Mixup (Zhang et al., 2018) and MixMatch (Berthelot et al., 2019) might be the best in terms of simplicity and efficiency. Thus, in this study, the authors chose Mixup and MixMatch as the base semi-supervised training algorithm. The integration of top-k pseudo labeling and Mixup/MixMatch semi-supervised training algorithm produces a powerful method. Indeed, the experiments demonstrate that it is robust to the number of labeled data, and exhibits fast convergence ability. Notably, using Wide Resnet (Zagoruyko & Komodakis, 2016) as the backbone, with the help of top-k pseudo labeling, the Mixup/MixMatch achieves 71.3%/90.10% top-1 test accuracy within 1000 epochs of training when there were only 25 labeled data per class (250 total labeled data) over CIFAR-10 dataset, respectively.

The main contributions of the paper are as follows:

1. The authors propose a new method named top-k pseudo labeling to generate pseudo labels by the classifier being trained. The method is easy to implement and can be integrated into any self-learning based semi-supervised learning methods.
2. To solve the problem of the choosing hyperparameter k, the authors propose dynamic schedule policies for k in top-k pseudo labeling method.
3. The authors propose a variant of top-k labeling for supervised learning named top-k regulation. Experiment results show that it improves the accuracy of various models on CIFAR-10/100 dataset.

BACKGROUND

Cotraining (Blum & Mitchell, 1998) may be the first attempt to classify unlabeled data with the help of enormous labeled data. In many machine learning contexts, unlabeled data are more readily available than labeled data. A relevant example is the problem of Web page classification. Classifying Web pages into categories requires much time to manually label hundreds of millions of unlabeled Web pages. The purpose of cotraining (Blum & Mitchell, 1998) is to take advantage of many unlabeled

Web pages as much as possible. Cotraining provides a framework based on principal component analysis for learning from both labeled and unlabeled data.

Mixup (Zhang et al., 2018) is in fact a data augmentation method that effectively helps alleviating the confirmation bias problem (Tarvainen & Valpola, 2017). Mixup trains a neural network on convex combinations of pairs of examples and their labels; then, it regularizes the neural network to favor simple linear behavior in-between training examples. The Mixup random selects labeled data (x_i, y_i) and unlabeled data (x_j, y_j), and constructs a virtual training example $\left(\lambda x_i + \left(1-\lambda\right)x_i, \lambda x_j + \left(1-\lambda\right)x_j\right)$, in which $\lambda \in \left[0,1\right]$. In this way, Mixup extends the training distribution by incorporating the prior knowledge that linear interpolations of feature vectors should lead to linear interpolations of the associated targets. Although the Mixup method is straightforward and introduces minimal computation overhead, it gave a new state-of-the-art performance on the CIFAR-10 and the CIFAR-100 dataset in 2018.

Mixup (Zhang et al., 2018) can be categorized into self-training and semi-supervised training algorithm, too. The idea of self-learning is to first train the classifier with labeled data and then to use the trained classifier to classify and get a predicted label for unlabeled data. Finally, Mixup combines labeled data and the data with the predicted label into a new training dataset. The classifier is re-trained with the new training dataset, and the procedure repeated until a satisfied classifier is got. This is why the method is called self-training.

The biggest problem of the vanilla self-training technique is that the classifier uses its own prediction to train itself. As a result, the mis-predicated labels will reinforce themselves, and then the classifier will overfit to incorrect labels. This is the so-called confirmation bias problem (Tarvainen and Valpola, 2017). The root cause of the problem is that the classifier is supervised by wrong labels, the classifier itself cannot reclaim the wrong labels, and this degenerates the model. Mixup (Zhang et al., 2018) is a good attempt to solve the problem, which demonstrates good generalization ability. Both MixMatch (Berthelot et al., 2019) and FixMatch (Sohn et al., 2020) extend Mixup method. MixMatch (Berthelot et al., 2019) guesses low-entropy labels for data-augmented unlabeled examples and mixes labeled data and unlabeled data using Mixup. MixMatch feeds N augmented unlabeled data into classifiers and uses the sharpened average of N prediction as pseudo label; this process is called label guessing or labeling.

FixMatch (Sohn et al., 2020) first generates pseudo-labels using the model's predictions on weakly augmented unlabeled images. For a given image, the pseudo-label is only retained if the model produces a high confidence prediction. The model is then trained to predict the pseudo label when fed with a strongly augmented version of the same image (Sohn et al., 2020). Briefly, FixMatch feeds weakly augmented and strongly augmented unlabeled data into the classifier, and gets the two predictions P_{weak} and P_{strong}. Then, FixMatch then clips the P_{weak} with predefined threshold and uses the clipped P_{weak} as the pseudo label. The model is trained to make its prediction on cross-entropy loss between P_{weak} and P_{strong}.

Self-Learning with Multi-Prototypes (Han et al., 2019) is a pseudo labeling method that contains two phases of training. The first phase is the training phase; the sum of the weighted loss of cross-entropy between prediction and noisy label and between prediction and corrected label is used as the total loss value. The second phase is the label correction phase; in this phase, the unlabeled data that randomly select the labeled images from each class are fed into the classifier, then the features of unlabeled images and features of labeled images are clustered and guide the correction of the noisy label to obtain a corrected label. The two phases are carried out iteratively.

Semi-Supervised Contrastive Learning (Zhang et al., 2022) is another pseudo label correction and generation method. It incorporates contrastive loss into semi-supervised learning by introducing a co-calibration mechanism to interchange predictions between the two branches. One branch uses cross-entropy loss for pseudo label generating, and the other branch uses the pseudo label to search for nearest neighbors for contrastive learning. The similarity embedding learned by the contrastive

branch is used in turn to adjust the pseudo label. Transudative Semi-Supervised Deep Learning (Shi et al., 2018) introduces confidence levels on unlabeled image samples to overcome unreliable label estimates on outliers and uncertain samples. It develops the Min-Max feature regularization that encourages deep convolutional neural networks to learn feature descriptors with better between-class separability and within-class compactness.

All the methods the authors briefly described in this section use a pseudo label that is generated and calibrated during training loops to train the model itself. Thus, how the pseudo label is generated for unlabeled data is important for the type of semi-supervised training algorithm.

MAIN FOCUS OF THE STUDY

In this study, the authors proposed a pseudo labeling method for semi-supervised learning, named top-k labeling. In a multi-class classification problem, SoftMax is always the last layer. In other words, SoftMax assigns decimal probabilities to each class in the classification problem. Those decimal probabilities are assumed to be summed to 1.0. This additional constraint helps the model converge more quickly than it otherwise would, and makes each decimal behave like likelihood of an image belonging to a particular class. As to the semi-supervised learning, how to choose appropriate pseudo labels for unlabeled data to guide the training process is very important. The choice of appropriate pseudo label is also called label guessing. A very direct label guessing method is to use the network prediction as pseudo label (Lee, 2013). In their work, Lee generated the pseudo labels after the warm-up training stage. Shi et al. (2018) used the network prediction as hard pseudo labels for unlabeled data, and they used K-nearest neighbors algorithm to calculate the confidence level of each sample to discount influences from those adverse samples. In recent works such as MixMatch, scholars have used average of the predictions of multiple augmented unlabeled data as pseudo label, which is a type of soft pseudo labels, too (Berthelot et al., 2019).

Guessing labels from model prediction are sometimes incorrect, and, in turn, the incorrect labels of unlabeled data reinforce the model to predict incorrect labels. Overfitting to incorrect pseudo labels predicted by the network is known as confirmation bias (Arazo et al., 2020; Tarvainen & Valpola, 2017). To help alleviate the confirmation bias problem, Berthelot et al. (2019) used the average of the predictions of multiple augmented unlabeled data as pseudo label, while Sohn et al. (2020) used cosine distance of unlabeled data to pre-selected labeled data as prior knowledge to reweighting the network prediction, and then used the reweighted prediction as pseudo label.

Top-k Labeling

In the classification task, the prediction output of the neural model is in the form of the SoftMax. Each element of the SoftMax output can be treated as the likelihood of the input to belong to each class. Normally, the argmax of the SoftMax of the logits is treated as the prediction. As an example, the authors assume that the prediction of the network is $h_\theta\left(x\right)$ in which θ represents the model parameters and x represents the input (labeled or unlabeled data). If the output of the model $h_\theta\left(x\right)$, is produced by the SoftMax layer, then $[y_0, y_1, \cdots, y_{C-1}] = h_\theta\left(x\right)$,in which C is the number of classes in the dataset. If the model is very confident that x belongs to class i, then y_i has the maximum value among $[y_0, y_1, \cdots, y_{C-1}]$. Ideally, y_i is near to 1.0, and the sum of reset elements is near to 0.

Firstly, the authors define the noise of guessed label for unlabeled data in the neural model output. As they previously stated, considering x is the unlabeled data, and y is the model prediction of x, then $[y_0, y_1, \cdots, y_{C-1}] = h_\theta\left(x\right)$. Typically, the model output is calibrated by the SoftMax layer, then $\sum_{i=0}^{C-1} y_i = 1.0$. If y_i is treated as the probability that x belongs to class i, and the ground truth label of x is y_{true} then the noise can be defined as Equation 1, in which p_i is the probability that y_i is the true label for x:

$$noise_{\theta} = \sum_{i \neq y_{true}} y_i \cdot x_i \tag{1}$$

Obviously, the noise from the guessed label will influence the convergence of the semi-supervised model, thus the lower the noise the higher the accuracy. For labeled data, the ground truth label has zero noise.

As the ground truth label of unlabeled data is unknown, the noise cannot be well defined as Equation 1 because p_i is unknown. The aim of the top-k labeling method is to reduce the label noise to facilitate the training of the semi-supervised model. The assumption is that the model may have a level of confidence on that fact that unlabeled data x do not belong to the subset of labels top_K, also known as $y_i \notin top_K$, where top_K is the set of index i so that y_i is the K top most value from $[y_0, y_1, \cdots, y_{C-1}]$, in which C is the number of classes and K (in upper case) is some specific value for hyperparameter k. Then the noise of guessed label can be separated into two parts, as Equation 2 and Equation 3 state:

$$noise_{\theta} = \sum_{i \in top_k} y_i \cdot x_i + \sum_{i \notin top_k} y_i \cdot x_i \tag{2}$$

$$noise_{\theta} = \frac{\sum_{i \in top_k} y_i \cdot p_i}{K} \sum_{i \in top_k} y_i + \frac{\sum_{i \notin top_k} y_i \cdot p_i}{K} \sum_{i \notin top_k} y_i \tag{3}$$

$\sum_{i \in top_K} p_i$ and $\sum_{i \in top_K} p_i$ can be estimated on a validation set as posterior in the context of supervised learning. However, in semi-supervised learning, Equation 3 is ill-formed. Taking semi-supervised training on CIFAR-10 as an example, as the training in progress of the top-k accuracy is increasing steadily, the top-9 accuracy of the model being trained will achieve a considerable level quickly; thus, for any unlabeled x there is a label y_i in which $i \notin top_9$. It is possible to conclude that y_i is not the true label for x, and thus remove it (i.e., set the corresponding elements to zero) from the guessed label y will reduce the noise. Based on this assumption, Equation 3 can be revised into Equation 4:

$$noise_{\theta}^{'}\left(x\right) \approx \frac{\sum_{i \in top_K} p_i}{K} \sum_{i \in top_K} y_i \leq noise_{\theta}\left(x\right) \tag{4}$$

In the context of semi-supervised learning, the ground truth labels of the unlabeled data are unknown, and the model $h_{\theta}\left(x\right)$ is trained by guessed labels. To help the model get optimal θ, if it is not possible to tell the optimizer which label is right, why not telling the optimizer which label is wrong? Based on this idea, the authors developed an algorithm named top-k labeling. The aim was to keep only the top-k elements of SoftMax prediction untouched and set others to zero when guessing labels for unlabeled data. Making full use of the top-k elements of the guessing label may help to alleviate confirmation bias problem by reducing the noise from the guessed label and may help to achieve higher prediction accuracy on unlabeled data. The top-k label guessing can be described as the algorithm in Figure 1.

For semi-supervised learning algorithms such as Mixup, the top-k label guessing can be applied directly. For semi-supervised learning algorithms with data augmentation, such as MixMatch, there are two ways to apply top-k label guessing. The first is to apply the top-k label guessing algorithm (as described in Figure 1) on the average of prediction of the augmented unlabeled data; the second is to apply the top-k label guessing algorithm to each of the prediction of the augmented unlabeled data firstly, and then use the average of the output of top-k label guessing algorithm as the guessed label. In this study, the authors used only the first method.

Choosing k

The k in top-k labeling is a hyperparameter. If k equals to the number of classes, the label generated by the top-k labeling degenerates to soft label. If k is fixed to 1, the top-k labeling generates a hard label. The choice of k may influence the convergence of the model. Choosing a bigger value for k implies the guessed pseudo label will include likelihoods of more possible classes and introduce more noise. Selecting a smaller value for k means the guessed pseudo label will include likelihoods to less possible classes but may opt out the true class from label. As a result, the authors empirically developed the following three kinds of methods for choosing the value for k for semi-supervised learning, and named these methods as schedule policies of k:

1. **Constant policy:** Using the same k during training. k can be chosen from 2 to C (C is the number of classes). Usually, the authors do not set k to 1 because the prediction of the model is incorrect in most cases, especially during the early stage of training. The constant policy is denoted as k_N, in which N is the value of k.
2. **Descendent policy:** k value monotonic descendent from high value to low value. The value of k is set to C at the beginning and decreases to 2 linearly till the end of training. This policy is denoted as p1, p5, and p6, in which k descendent linearly in p1 and descendent like a parabola in p5 and p6.
3. **Repeated descendent policy:** The training progress is divided into multiple stages, k descendent from high value to low value inside each stage. k is set to C at the beginning and decrease to 2 linearly till the end of each stage. The policy is denoted as p3.

The descendent policy of k is intuitive. At the beginning of the training, the model cannot tell which class is incorrect with confidence, so a bigger k is appropriate. As the training progresses, the model gets more accurate and a smaller k makes pseudo label closer to the real label. Figure 2 shows policies p1, p3, p5, and p6.

Semi-Supervised Training With Top-k Labeling

The top-k labeling method can be directly integrated into any self-learning based semi-supervised training algorithm. In this study, the authors applied the top-k labeling method to Mixup and MixMatch. MixMatch is based on Mixup, and MixMatch exhibits state-of-the-art performance by a large margin across many datasets and labeled data amounts. In the experiments, with Wide Resnet 28x2 as the backbone and only 250 labeled data on CIFAR-10 dataset, Mixup and MixMatch algorithms get the accuracy of 71.9% and 90.52% respectively with the help of top-k labeling. The section Experiments provides relevant details.

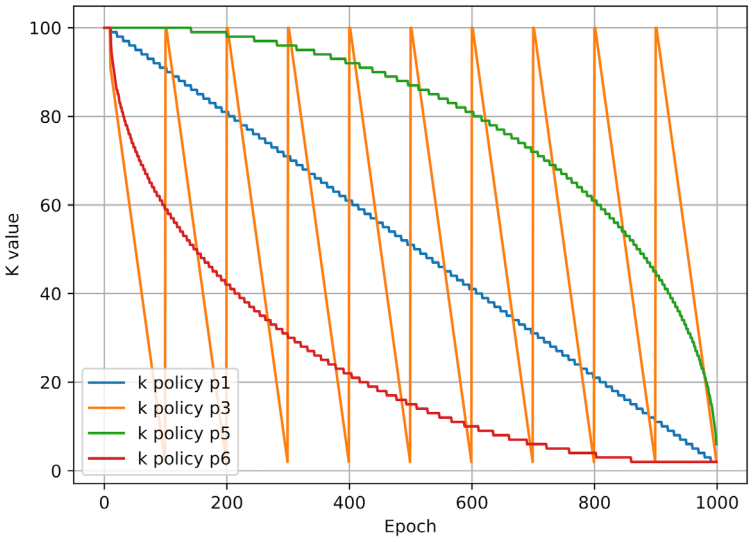
The MixMatch algorithm is a powerful method and presents the current state-of-the-art performance. In the experiments, the authors simply applied the top-k labeling method to MixMatch to generate pseudo labels from the model outputs. The Mixup algorithm with top-k labeling (Figure 3) describes

Figure 1.
Top-k label guessing algorithm

Require: $K < C$ ▷ C is the number of classes
Require: $y = h_\theta(x)$ ▷ y is the prediction of unlabeled data x
$i_0, i_1, \ldots, i_{K-1} = numpy.argpartition(y, -k, 1)$
for $j = 0$ to k **do**
$y_{i_j} \leftarrow 0$
end for

for $j = 0$ to $C - 1$ **do**
$y_j \leftarrow \frac{y_j}{\sum_{k=0}^{C-1} y_k}$
end for

Figure 2.
Example of k scheduling in top-k labeling, assuming the number of classes is 100 and 1000 epochs of training



a training step of model h_θ with labeled data $\left(x_l, y_l\right)$ and unlabeled data $\left(\hat{x}_u, \_\right)$. At the beginning of the training step, unlabeled data $\hat{x}_u$ are randomly augmented twice, the result is $\hat{x}_{u1}$ and $\hat{x}_{u2}$, respectively.

Top-k Regulation in Supervised Learning

In supervised learning, the model is trained by the data with ground truth labels, and there is no need to use guessed labels as target. In this section, the authors introduce a variant of top-k labeling for supervised learning named top-k regulation. The top-k regulation for supervised learning differs from the top-k labeling for semi-supervised learning, but it is still simple enough to implement, that is, simply leaving top-k elements in the SoftMax output of the model along and setting the other elements to zero.

Figure 3.
Mixup with top-k labeling algorithm

Require: $K < C$ ▷ C is the number of classes
Require: α ▷ α is a hyper paramter of Mixup
Require: $\lambda > 0$ ▷ λ hyperparameter
Require: (x_l, y_l), $(\hat{x}_u, \_)$ ▷ the labeled and unlabeled data respectively
Require: h_θ ▷ the model to be optimized
Require: $h_{\theta_{best}}$ ▷ model with best loss value on unlabeled set
$\hat{y}_u, \hat{y}_{u1}, \hat{y}_{u2} \leftarrow h_{\theta_{best}}(\hat{x}_u, \hat{x}_{u1}, \hat{x}_{u2})$
$\bar{y}_u \leftarrow (\hat{y}_u + \hat{y}_{u1} + \hat{y}_{u2})/3$
$\hat{y}_u \leftarrow top\_k\_labeling(\bar{y}_u)$
$\lambda \leftarrow random.beta(\alpha, \alpha)$
$x_{mixup} \leftarrow \lambda * x_l + (1 - \lambda) * x_u$
$l_x \leftarrow L_{MSE}(h_\theta(x_{mixup}), y_l)$
$l_u \leftarrow L_{MSE}(h_\theta(x_{mixup}), \hat{y}_u)$
$loss_{mixup} \leftarrow \lambda * l_x + (1 - \lambda) * l_u$
$loss_{mixup}.backwards()$

The assumption is $y = h_\theta\left(x\right)$, in which x is the data fed into the neural model $h_\theta\left(x\right)$, y is the label of x, $\hat{y} = \left[\hat{y}_0, \hat{y}_1, \cdots, \hat{y}_{C=1}\right]$ is the SoftMax output and C is length of target vector. The top-k regulation method is to set $\hat{y}_i$ to 0 for all $i \notin argmax_K(\hat{y})$ and then normalize $\hat{y}$ as $\hat{y} = \frac{\hat{y}}{\sum_{i=0}^{C-1}\hat{y}_i}$. The loss value is then computed by $loss\left(y, \hat{y}\right)$ but not $loss\left(y, h_\theta\left(x\right)\right)$, in which *loss* is the loss function.

In supervised learning, k can be chosen during the training progress according to the top-k accuracy on validation or testing set. This policy is named as p0. For policy p0, k is chosen as the lowest k that makes top-k accuracy on test set greater or equal to the threshold at the end of each epoch. Obviously, the threshold can be a hyperparameter, too. For simplicity, the authors fixed the threshold value to 0.99, in this study.

In this section, the authors evaluate the top-k labeling method with MixMatch and Mixup algorithm on CIFAR-10/100 datasets (Krizhevsky, 2009), and top-k regulation with various base models on CIFAR-10/100 datasets.

For top-k labeling with MixMatch and Mixup, the authors chose Wide Resnet 28x2 as the backbone for CIFAR-10 dataset, and Wide Resnet 28x4 for CIFAR-100 dataset. When training the model with MixMatch, λ_u and α are set to 75 and 0.5, for CIFAR-10, and to 150 and 0.75 for CIFAR-100, respectively. When training the model with Mixup, α is set to 2.0, for CIFAR-10 dataset, and set to 0.5, for CIFAR-100 dataset. The mean squared error loss is used in all experiments for both labeled data and unlabeled data with Mixup and MixMatch algorithm.

For supervised learning with top-k regulation experiments, the authors tested various base models including Wide Resnet 28x2 and Wide Resnet 28x4. Those models are trained by Adam optimizer and cross-entropy loss, and the learning rates are fixed to 0.002 in all experiments.

The hardware and software configuration in the experiments are abbreviated as follows: Ubuntu Linux 18.04 LTS/amd64, Nvidia RTX 2080Ti GPU, CUDA 11.2, PyTorch 1.10.

Dataset

To demonstrate the efficiency of the proposed method, the authors provided comprehensive comparison results on CIFAR-10/100 datasets. The CIFAR-10 and CIFAR-100 dataset are two sets of images with reliable labels. The CIFAR-10 dataset has 6000 examples of each of 10 classes, and the CIFAR-100 dataset has 600 examples of each of 100 non-overlapping categories. As the authors mentioned, both CIFAR-10 and CIFAR-100 dataset have 60000 images in total. In training, both labeled and unlabeled data are randomly cropped and randomly flipped (horizontally and vertically).

Experiment Setup

In the experiments, each dataset is split into a training set (labeled), a validation set (unlabeled), and a testing set. The number of labeled data are selected from 250, 500, 1000, 2000, and 4000. The labeled data and unlabeled data for training are randomly chosen from the training split of the dataset. An example is CIFAR-10 dataset and 250 labeled data, which are 250 labeled images and 49,250 unlabeled data for training, 500 labeled data from the training split of the dataset for validation, and 10000 labeled data from the testing split of the dataset for testing. Also, the authors applied different schedule policies of k in the experiments. As to k, k5 means k is a constant of value 5 during the training, p3 means k is scheduled by policy 3 during training, and so on. Finally, the authors presented in the Figures and Tables only top-1 accuracy on test split of the dataset.

Experiments on CIFAR-10

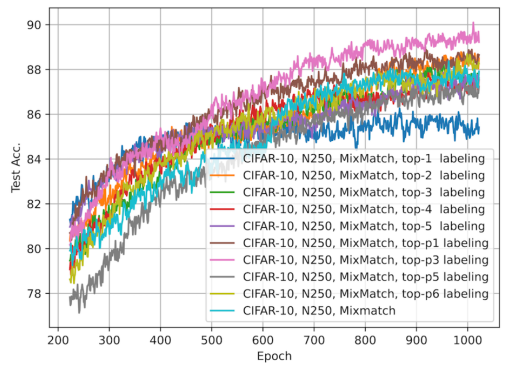
The authors used Wide Resnet 28x2 as the base model on CIFAR-10 dataset to evaluate the accuracy of the model with top-k labeling with different k values and schedule policies. The value of k varies from 2

to 3, 4, 5 if k is a constant, and the schedule policy of k can be p1, p3, p5 and p6 if k is a scheduled. The experiments on CIFAR-10 dataset with different numbers of labeled data are shown in Fig 4 and Table 1.

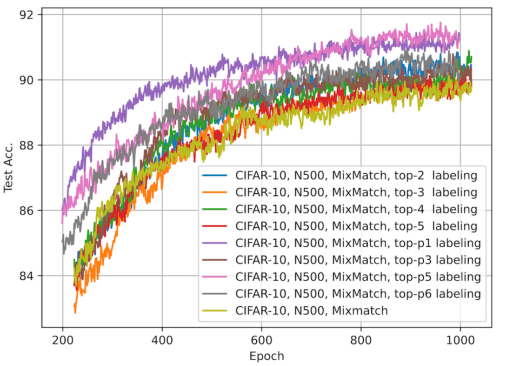
As Figure 4(a) illustrates, the MixMatch algorithm with top-1 labeling achieved the worst accuracy among all the configurations. The Mixup algorithm with top-p3 and top-2 are the best and the second best, respectively. The results are not unexpected, as in semi-supervised learning labels guessed during the early stage of the training are mostly incorrect; setting k to 1 will obviously reinforce the confirmation bias problem. MixMatch with the top-p3 labeling method showed an excellent accuracy of 90.1%, which is better than MixMatch without the top-k labeling by almost 2%.

Figure 4(b), Figure 4(c), and Figure 4(d) are the test accuracy curves of Wide Resnet 28x2 model trained by MixMatch with top-k labeling; the number of labeled data are 500, 1000, and 2000 respectively. As Figure 4 and Table 1 show, top-p3 labeling gets the best accuracy when the number of labeled data is 250 and 4000, top-p5 labeling gets best accuracy when the number of labeled data are 500 and 2000, and top-p6 gets the best accuracy when the number of labeled data is 2000. In all experiments, MixMatch with top-k labeling outperforms original MixMatch by different levels. Also, the experiment results in Table 1 suggest that the accuracy improvement by top-k labeling method decreases as the number of labeled data increase, which means the top-k labeling performs well, especially when guessed labels have more noises. It is clear that the less the labeled data, the more the noise in the guessed label.

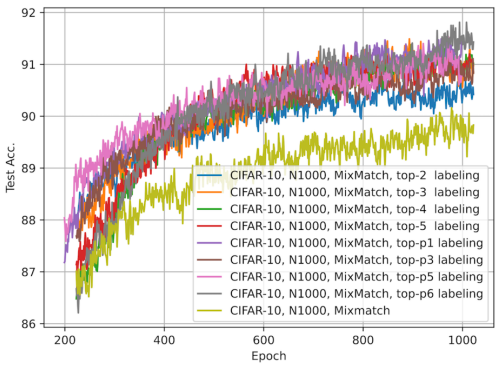
Figure 4.
Accuracy Obtained on CIFAR-10 Test Set, Model Trained by MixMatch with Top-k Labeling with 250, 500, 1000, and 2000 Labeled Data, Respectively



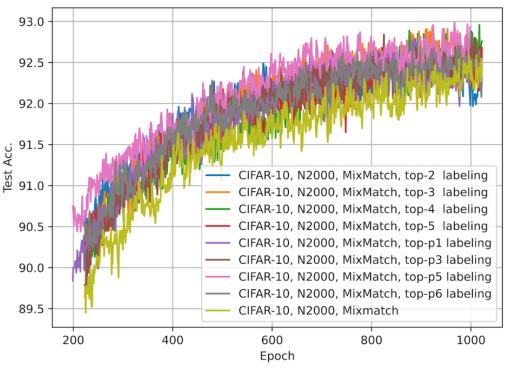
(**a**) $N_{labeled} = 250$.



(**b**) $N_{labeled} = 500$.



(**c**) $N_{labeled} = 1000$.



(**d**) $N_{labeled} = 2000$.

Table 1.
Accuracy of Wide Resnet 28x2 Model Trained by MixMatch Algorithm With and Without Top-k Labeling After 1000 Epochs of Training on CIFAR-10 Dataset, Learning Rate Fixed to 0.0002, and λ_u = 75.0

$N_{labeled}$	MixMatch with top-k labeling								MixMatch
	k=2	k=3	k=4	k=5	k=p1	k=p3	k=p5	k=p6	
250	88.78	88.23	87.84	87.88	89.10	90.10*	87.45	88.69	88.47
500	90.85	90.30	90.89	90.21	91.45	90.79	91.76*	90.93	90.12
1000	90.86	91.49	91.38	91.32	91.49	91.11	91.54	91.81*	90.81
2000	92.79	92.94	92.96	92.70	92.79	92.84	92.99*	92.71	92.59
4000	93.97	93.88	93.88	94.07	93.82	94.09*	93.86	94.02	93.86

Note: * best accuracy in each row. k is chosen from 2, 3, 4, 5, p1, p3, p5 and p6, the number of labeled data is chosen from 250, 500, 1000, 2000, and 4000.

Considering MixMatch already presents state-of-the-art semi-supervised performance on CIFAR-10 dataset, the authors chose to apply the top-k labeling method for the Mixup algorithm, which is less accurate than many of the semi-supervised training algorithms today. The authors designed experiments to show the accuracy improved by the top-k labeling method, but not to outperform the state-of-the-art performance; so, for each combination of the k and number of labeled data, the training only takes 300 epochs. Figure 5 and Table 2 show the experiment results.

The top-k labeling contributes almost 20% percent to the accuracy when there are 250, 500, and 1000 labeled data, and 5.5% when there are 2000 labeled data for training (Table 2 and Figure 5). In all the experiments, Mixup with top-k labeling outperforms the original Mixup algorithm (without top-k labeling).

Based on the experiment results in this section, the authors concluded that top-k labeling effectively improves the test accuracy on Wide Resnet 28x2 trained by MixMatch and Mixup algorithm. When there are fewer labeled data for training, the model is less accurate, and then the guessed label (generated by model prediction) contains more noise. Obviously, the top-k labeling method helps much in this situation.

Experiments on CIFAR-100

Unlike the CIFAR-10 dataset, the CIFAR-100 dataset has 100 classes, and CIFAR-100 dataset has fewer data per class than CIFAR-10 dataset. Thus, the authors used Wide Resnet 28x4 (5.9 million parameters) as the backbone model. The experiment settings are the same as in the previous section, and in the experiments the authors used 2000, 3000, 4000, and 10000 labeled data for training. k was chosen from 2, 5, 10, p1, p3, p5, and p6. Table 3 and Figure 6 show the experiment results using MixMatch training algorithm, while Table 4 and Figure 7 show the experiment results using Mixup training algorithm. The results show that for both MixMatch and Mixup training algorithm and a different number of labeled training data, the semi-supervised training algorithm with top-k labeling outperforms the original algorithm at different levels of improvement in many cases.

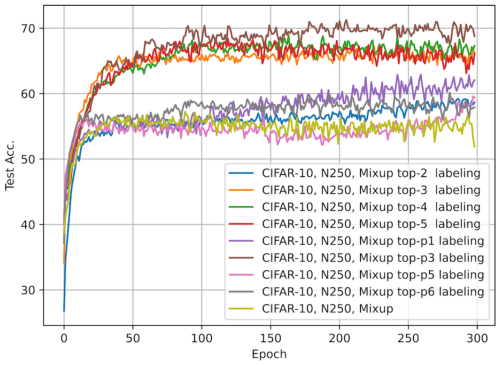
The experiment results above confirm that the top-k labeling performs better when there are less labeled data for training, or, briefly, when there are more noises in the guessed (pseudo) labels.

Top-k Regulation in Supervised Learning

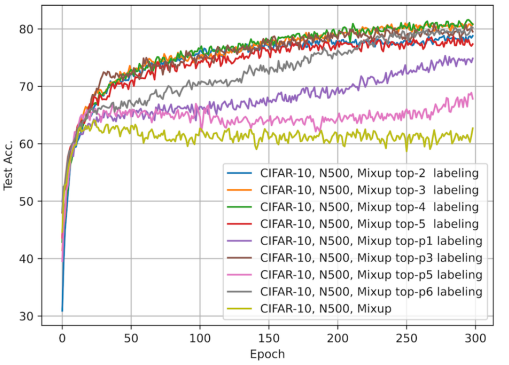
The experiments in this section demonstrated the efficiency of the top-k regulation in supervised learning. Table 5 and Table 6 show the results on Wide Resnet 28x2 model with CIFAR-10 and with CIFAR-100 datasets, respectively.

As Table 5 shows, in most cases top-k regulation will not bring negative impact to the convergency of the training. On the contrary, the accuracy is improved at different level respectively on different

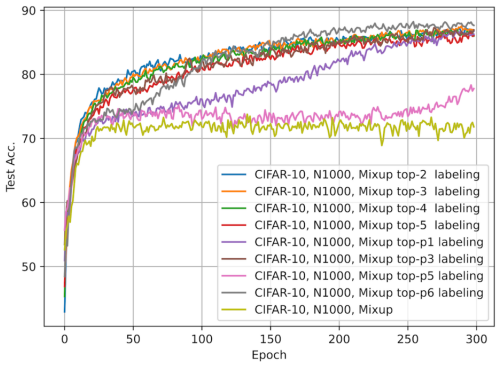
Figure 5.
Accuracy Obtained on CIFAR-10 Test Set, Model Trained by Mixup with Top-k Labeling With 250, 500, 1000, 2000 Labeled Data, Respectively



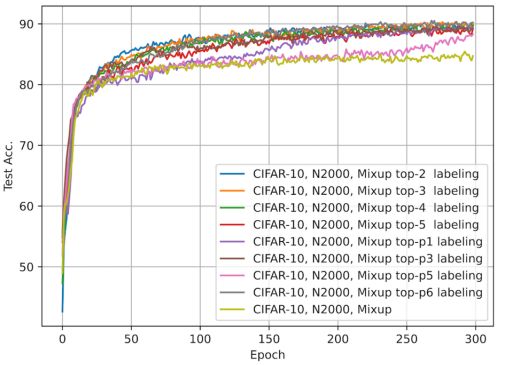
(**a**) $N_{labeled} = 250$.



(**b**) $N_{labeled} = 500$.



(**c**) $N_{labeled} = 1000$.



(**d**) $N_{labeled} = 2000$.

Table 2.
Accuracy of Wide Resnet 28x2 Model Trained by Mixup Algorithm With and Without Top-k Labeling After 300 Epochs of Training on CIFAR-10 Dataset, Learning Rate Fixed to 0.002

$N_{labeled}$	Mixup with top-k labeling								Mixup
	k=2	k=3	k=4	k=5	k=p1	k=p3	k=p5	k=p6	
250	59.23	67.37	68.91	68.40	62.91	71.28*	59.60	60.18	57.18
500	78.82	80.99	81.58*	78.75	75.32	80.86	68.89	80.38	64.06
1000	87.32	87.55	86.97	86.64	86.68	87.28	78.34	88.23*	73.77
2000	90.23	90.31	89.74	89.26	89.71	89.58	88.28	90.54*	85.45
4000	91.73	91.53	91.44	91.77	91.13	91.53	91.12	91.90*	90.64

Note: * best accuracy in each row. k is chosen from 2, 3, 4, 5, p1, p3, p5, and p6; the number of labeled data is chosen from 250, 500, 1000, 2000, and 4000.

k. Although the improvement is small, it does prove that top-k labeling/regulation method is effective in supervised training. Besides, comparing the result between Table 2 and Table 5, the authors found

Table 3.
Accuracy of Wide Resnet 28x4 Model Trained by MixMatch Algorithm With and Without Top-k Labeling After 200 Epochs of Training on CIFAR-100 Dataset, Learning Rate Fixed to 0.002, α set to 0.75, and λ_u set to 150

$N_{labeled}$	MixMatch with top-k labeling							MixMatch
	k=2	k=5	k=10	k=p1	k=p3	k=p5	k=p6	
2000	51.96	51.74	51.14	50.45	52.21	52.25*	49.88	50.77
3000	57.10	59.25*	58.03	56.82	57.87	57.66	55.59	57.65
4000	61.49	61.46	61.61	59.67	60.84	60.27	59.67	61.86*
10000	71.02	70.64	69.97	70.09	69.53	68.80	70.33	71.41*

Note: * best accuracy in each row. k is chosen from 2, 5, 10 p1, p3, p5 and p6, the number of labeled data is chosen from 2000, 3000, 4000, 10000.

Table 4.
Accuracy of Wide Resnet 28x4 Model Trained by Mixup Algorithm With and Without Top-k Labeling After 500 Epochs of Training on CIFAR-100 Dataset, Learning Rate Fixed to 0.002

$N_{labeled}$	Mixup with top-k labeling							Mixup
	k=2	k=5	k=10	k=p1	k=p3	k=p5	k=p6	
2000	22.33	40.99	44.00	45.92	48.74	41.71	48.95*	44.35
3000	38.40	49.98	50.24	55.22*	54.68	54.58	54.41	54.40
4000	40.46	54.33	56.75	58.19*	56.77	57.36	57.94	57.23
10000	63.51	63.74	65.12	66.71*	65.27	66.57	66.70	65.39

Note: * best accuracy in each row. k is chosen from 2, 5, 10, p1, p3, p5 and p6, the number of labeled data is chosen from 4000, 10000.

that the Mixup semi-supervised training algorithm with top-k labeling achieves the same level of accuracy as supervised training if the number of labeled data is greater or equal to 4000.

The results in Table 5 also indicate that Wide Resnet 28x2 fits CIFAR-10 dataset well with or without top-k regulation. For datasets with a smaller number of classes and many samples per class such as CIFAR-10, the prediction by Wide Resnet 28x2 model has little noise, which is why top-k regulation helps little. To demonstrate how the top-k regulation contributes to the fitting of the model, the authors conducted a series of experiments on the CIFAR-100 dataset and various base models.

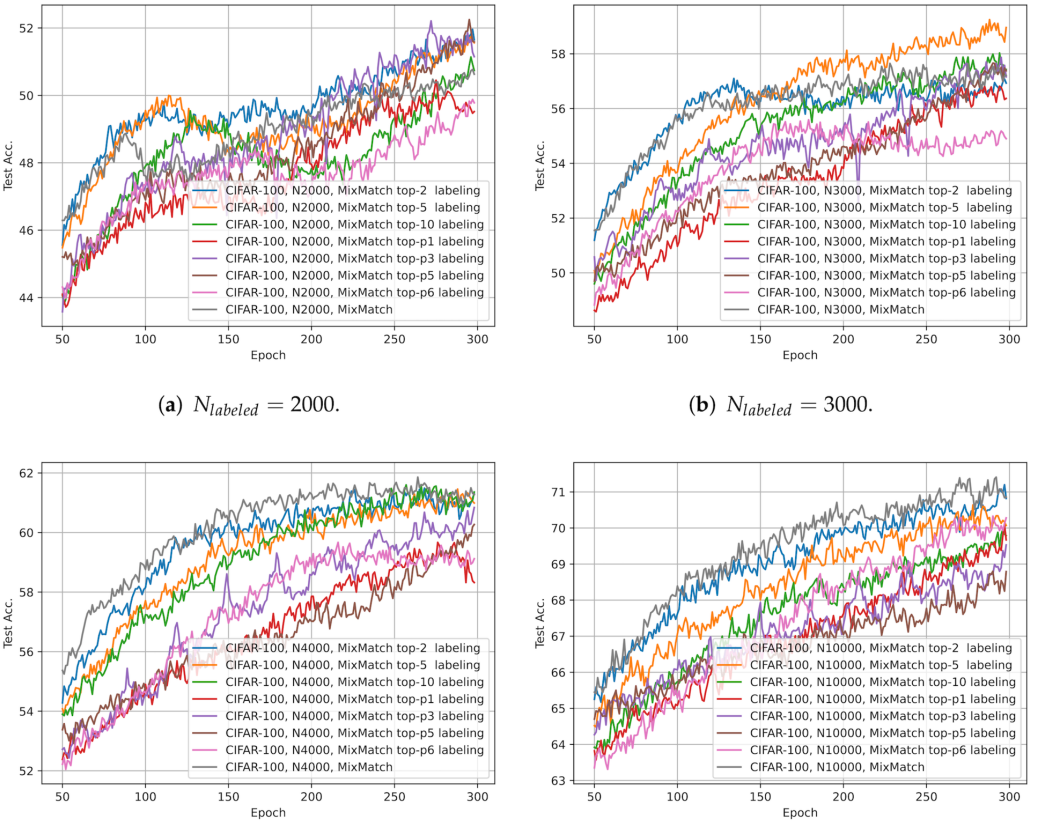
The results in Table 6 and Figure 8 and in Table 7 and Figure 9 show that top-k regulation with a smaller k value brings negative impact on the convergence of the training. This is due to the slow convergency of the base model on the CIFAR-100 dataset. If the model cannot converge to a reasonable accuracy quickly, top-k regulation with the smaller k will introduce extra noise in predictions.

On the other side, top-k with p0, p1, p3, p5, and p6 behaves very well. The nature of the abovementioned policies is starting k at 100 and decreasing as the pace of training epochs. These k policies mimic the warm-up process indeed, and finally achieve higher accuracy (Table 6 and Table 7). Figure 8 and Figure 9 highlight the top-p0 and top-p1 regulations exhibit strong regulation effects on model prediction; thus, this allows to observe a significant relative improvement in test accuracy, and to conclude that the top-k regulation reduces the noise effectively from model prediction.

Table 6 and Table 7 show that can be found that top-p0 and top-p1 are better than the other k policies. Thus, the authors tried to examine how the top-k regulation would do if the model was trained for more epochs (Table 8). The experiment evidence that the top-k regulation effectively improves the performance of both the smaller model and the larger model.

The authors conducted further experiments on MobileNet (Sandler et al., 2018), DenseNet (Huang et al., 2017), GoogleNet (Szegedy et al., 2015), and ResNet (He et al., 2016). They aimed

Figure 6.
Accuracy Obtained on CIFAR-100 Test Set, Model Trained by MixMatch With Top-k Labeling With 2000, 3000, 4000, and 10000 Labeled Data, Respectively



(**a**) $N_{labeled} = 2000$.

(**b**) $N_{labeled} = 3000$.

(**c**) $N_{labeled} = 4000$.

(**d**) $N_{labeled} = 10000$.

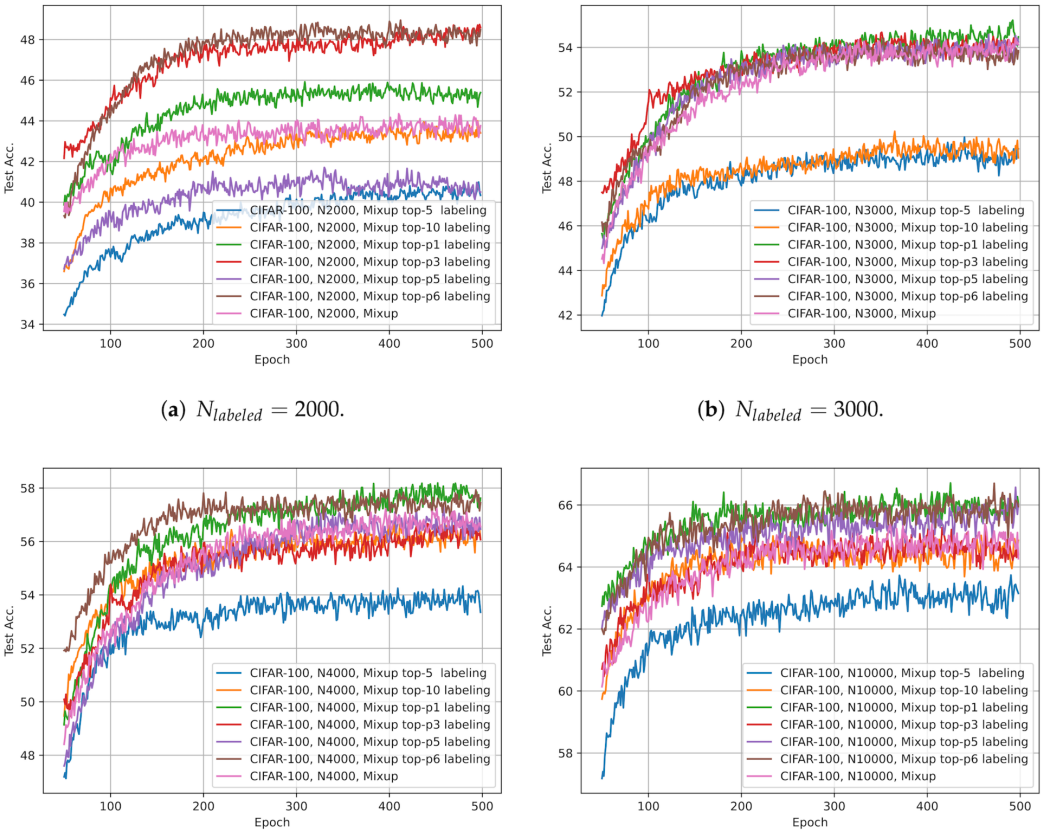
to test how top-k regulation helps the model to converge, but not to achieve a state-of-the-art result. Table 8 show the experiment results. For each of the base models, top-p0, top-p1, top-p3, top-p5, and top-p6 are deployed. The results show that top-k regulation outperforms the base model much on test accuracy, from 5% to 10%, respectively.

The experiments in this section demonstrate that properly selecting k value of top-k regulation in supervised learning will help the model to get higher accuracy by reducing prediction noise. Typically, top-k regulation with scheduled k value, such as p0 and p1, consistently performs well as a result of the warm-up process in the policy. The constant k value is not a good choice if the model tends to under-fit. Finally, k as a hyperparameter needs to be carefully selected to make the model fitting and generalization better.

CONCLUSION

In this study, the authors identified the noise inside pseudo label as a primary obstacle preventing self-learning based semi-supervised learning to get higher accuracy on unlabeled data, and then introduced top-k labeling method to generate pseudo labels for self-learning based semi-supervised learning. The method is easy to implement and easily integrated into any self-learning algorithms.

Figure 7.
Accuracy Obtained on CIFAR-100 Test Set, Model Trained by Mixup With Top-k Labeling With 2000, 3000, 4000, and 10000 Labeled Data, Respectively



(**a**) $N_{labeled} = 2000$.

(**b**) $N_{labeled} = 3000$.

(**c**) $N_{labeled} = 4000$.

(**d**) $N_{labeled} = 10000$.

Table 5.
Accuracy of Wide Resnet 28x2 Model Trained in Supervised Manner With and Without Top-k Labeling After 300 Epochs of Training on CIFAR-10 Dataset, With Learning Rate Fixed to 0.002

k	2	3	4	5	6	7	8	9	p0	p1	p3	p5	p6	n/a
Acc.	93.76	93.71	93.66	93.69	93.78	93.62	93.74	93.63	93.78	93.63	93.71	93.91*	93.58	93.64

Note: * best accuracy. k is chosen from 2 to 9, p0, p1, p3, p5 and p6, the best accuracy is in bold face. “n/a” means not applying the top-k regulation. The accuracy is computed on CIFAR-10 test set

Table 6.
Accuracy of Wide Resnet 28x2 Model Trained in Supervised Manner With and Without Top-k Labeling After 300 Epochs of Training on CIFAR-100 Dataset, With Learning Rate Fixed to 0.002

k	2	3	4	5	6	7	8	9	p0	p1	p3	p5	p6	n/a
Acc.	38.53	36.62	38.29	34.45	35.80	33.81	37.29	39.73	65.40	70.22	67.47	69.97*	69.53	63.55

Note: * best accuracy. k is chosen from 2 to 9, p0, p1, p3, p5 and p6, the best accuracy is in bold face. “n/a” means not applying the top-k regulation. The accuracy is computed on CIFAR-100 test set.

Figure 8.
Test Accuracy Obtained on Wide Resnet 28x2 and CIFAR-100 Dataset in Supervised Manner With and Without Top-k Regulation

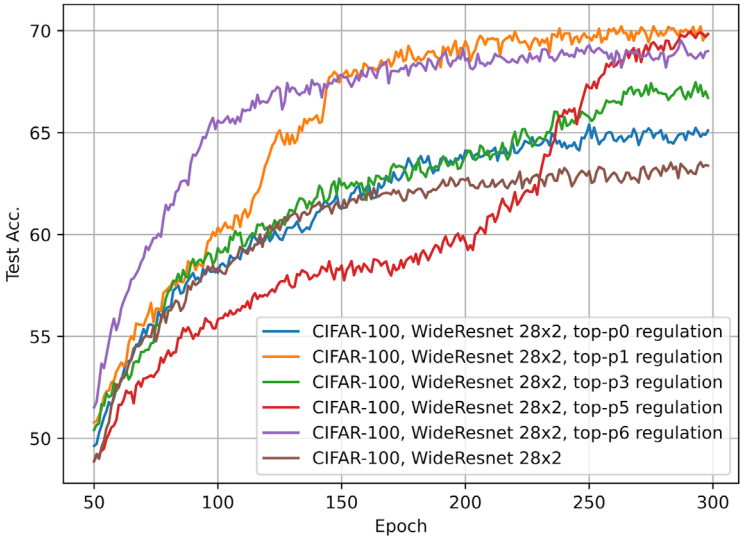


Table 7.
Accuracy of Wide Resnet 28x4 Model Trained in Supervised Manner With and Without Top-k Labeling After 300 Epochs of Training on CIFAR-100 Dataset, With Learning Rate Fixed to 0.002

k	2	3	4	5	6	7	8	9	p0	p1	p3	p5	p6	n/a
Acc.	43.69	45.13	37.99	36.76	41.69	41.01	44.68	42.11	74.69*	73.11	71.52	72.22	70.17	68.71

Note: * best accuracy. k is chosen from 2 to 9, p0, p1, p3, p5 and p6, the best accuracy is in bold face. 'n/a' means not applying the top-k regulation to neural model output. The accuracy is computed on CIFAR-100 test set.

Figure 9.
Test Accuracy Obtained on Wide Resnet 28x4 and CIFAR-100 Dataset in Supervised Manner With and Without Top-k Regulation

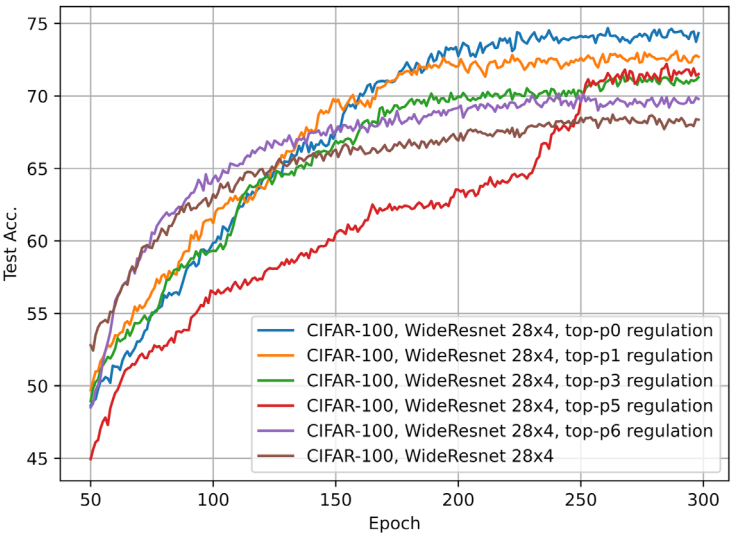


Table 8.
Accuracy of Various Base Model Trained in Supervised Manner With and Without Top-k Regulation After 1000 Epochs of Training on CIFAR-100 Dataset, With Learning Rate Fixed to 0.002

Base Model	Base Model With Top-k Regulation					n/a
	k=p0	k=p1	k=p3	k=p5	k=p6	
WideResNet 28x4	76.04	76.86*	76.05	75.43	74.59	71.31
WideResNet 28x8	77.94*	77.65	77.22	77.77	75.60	66.69
MobileNet v2	47.10	59.94*	55.89	55.14	54.06	50.53
ResNet 18	73.64*	72.54	72.69	72.33	73.50	62.68
GoogleNet	65.64	68.18	67.99	68.40*	66.43	59.68
DenseNet 121	62.49	54.50	61.58	65.06*	56.17	56.36
DenseNet 161	62.35	65.80	65.08	64.84	63.78	66.13*

Note: * best accuracy in each row. k is chosen from p0, p1, p3, p5 and p6, the best accuracy in each row is in bold face. "n/a" means not applying the top-k regulation to neural model output. The accuracy is computed on CIFAR-100 test set.

The experiment results indicate that the method did help the self-learning based semi-supervised algorithm to get better accuracy on unlabeled data at different level. Also, the authors found that the top-k regulation, which is a variant of top-k labeling, can help the model generalize better and get higher accuracy in supervised learning.

One major limitation of this study is that the authors did not find out how to choose k in top-k labeling effectively. The choice of hyperparameter k is empirical, so many rounds of experiment may be necessary to find a good k. It would be interesting to further investigate how to choose k automatically during the training process. Lastly, the authors would like to conduct more experiments on various datasets to further investigate the efficiency of the top-k labeling method in the future.

CONFLICT OF INTEREST

The authors of this publication declare there is no conflict of interest.

FUNDING AGENCY

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors. Funding for this research was covered by the authors of the article.

REFERENCES

Arazo, E., Ortego, D., Albert, P., O'Connor, N. E., & McGuinness, K. (2020). Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). IEEE. doi:10.1109/IJCNN48605.2020.9207304

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., & Raffel, C. A. (2019). Mixmatch: A holistic approach to semi-supervised learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems* (pp.5049-5059). Academic Press.

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory* (pp. 92-100). doi:10.1145/279943.279962

Han, J., Luo, P., & Wang, X. (2019). Deep self-learning from noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 5138-5147). Academic Press.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770-778). IEEE.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700-4708). IEEE.

Kingma, D. P., Mohamed, S., Jimenez Rezende, D., & Welling, M. (2014). Semi-supervised learning with deep generative models. *Advances in Neural Information Processing Systems*, 27.

Krizhevsky, A. (2009). *Learning multiple layers of features from tiny images* [Unpublished master's thesis]. University of Tront.

Lee, D. H. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML* (Vol. 3, No. 2, p. 896). Academic Press.

Pu, Y., Gan, Z., Henao, R., Yuan, X., Li, C., Stevens, A., & Carin, L. (2016). Variational autoencoder for deep learning of images, labels, and captions. *Advances in Neural Information Processing Systems*, 29.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4510-4520). IEEE.

Shi, W., Gong, Y., Ding, C., Tao, Z. M., & Zheng, N. (2018). Transductive semi-supervised deep learning using min-max features. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 299-315). doi:10.1007/978-3-030-01228-1_19

Sindhwani, V., & Rosenberg, D. S. (2008). An RKHS for multi-view learning and manifold co-regularization. In *Proceedings of the 25th International Conference on Machine Learning* (pp. 976-983). doi:10.1145/1390156.1390279

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., & Li, C. L. et al. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, *33*, 596–608.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., & Rabinovich, A. et al. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1-9). IEEE.

Tarvainen, A., & Valpola, H. (2017). Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in Neural Information Processing Systems*, 30.

Xie, Q., Luong, M. T., Hovy, E., & Le, Q. V. (2020). Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10687-10698). doi:10.1109/CVPR42600.2020.01070

Zagoruyko, S., & Komodakis, N. (2016). Wide Residual Networks. In *Proceedings of the British Machine Vision Conference 2016*. British Machine Vision Association. doi:10.5244/C.30.87

Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2018). Mixup: Beyond empirical risk minimization. *Proceedings of the International Conference on Learning Representations*.

Zhang, Y., Zhang, X., Li, J., Qiu, R., Xu, H., & Tian, Q. (2022). Semi-supervised contrastive learning with similarity co-calibration. *IEEE Transactions on Multimedia*.

Zhu, X. (2005). *Semi-supervised learning literature survey*. http://digital.library.wisc.edu/1793/60444

Yi Jiang is currently a lecturer in school of measurement control technology and communication engineering in Harbin University of Science and Technology, China. He received the M.S. degree in Computer Science and Technology from Harbin University of Science and Technology, China, in 2002, and Ph.D. degree in Computer Science and Technology from Shanghai Jiao Tong University, China, in 2007. His research interests include wireless sensor networks, embedded system, computer vision, deep learning, and big data.

Hui Sun is currently an Associate Professor of measurement and control technology and communication engineering in Harbin Universityof Science and Technology, China. He received the M.S. degree in Detection Technology and Automatic Equipment from Harbin University of Science and Technology, China, in 2004, and Ph.D. degree in Precision Instruments and Machinery from Harbin University of Science and Technology, China, in 2021. His research interests include sensor technology, measurement and control technology, and precision instrument design.