

Article

Not peer-reviewed version

ReliaMatch: Semi-supervised Classification with Reliable Match

[Tao Jiang](#) , Luyao Chen , Wanqing Chen , Wenjuan Meng , [Peihan Qi](#) *

Posted Date: 30 June 2023

doi: 10.20944/preprints202306.2209.v1

Keywords: Deep learning; Semi-supervised learning; Pseudo labels; Classification; Reliable Match



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

ReliaMatch: Semi-Supervised Classification with Reliable Match

Tao Jiang ¹, Luyao Chen ¹, Wanqing Chen ¹, Wenjuan Meng ² and Peihan Qi ^{3,*}

¹ School of Cyber Engineering, Xidian University, Xi'an 710126, China; taojiang@xidian.edu.cn; {wqchen, luyaochen}@stu.xidian.edu.cn

² College of Information Engineering, Northwest A&F University, Yangling, Shaanxi 712100, China; wjmeng@nwsuaf.edu.cn

³ State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China; phqi@xidian.edu.cn

* Correspondence: phqi@xidian.edu.cn

Abstract: Deep learning has been widely used in various tasks such as computer vision, natural language processing, and predictive analysis, recommendation systems in the past decade. However, practical scenarios often lack labeled data, posing challenges for traditional supervised methods. Semi-supervised classification methods address this by leveraging both labeled and unlabeled data to enhance model performance, but they face challenges in effectively utilizing unlabeled data and distinguishing reliable information from unreliable sources. This paper introduces ReliaMatch, a semi-supervised classification method that addresses these challenges by using a confidence threshold. It incorporates a curriculum learning stage, feature filtering, and pseudo-label filtering to improve classification accuracy and reliability. The feature filtering module eliminates ambiguous semantic features by comparing labeled and unlabeled data in the feature space. The pseudo-label filtering module removes unreliable pseudo-labels with low confidence, enhancing algorithm reliability. ReliaMatch employs a curriculum learning training mode, gradually increasing training dataset difficulty by combining selected samples and pseudo-labels with labeled data. This supervised approach enhances classification performance. Experimental results show that ReliaMatch effectively overcomes challenges associated with the underutilization of unlabeled data and the introduction of error information, outperforming the pseudo-label strategy in semi-supervised classification.

Keywords: deep learning; semi-supervised learning; pseudo labels; classification; reliable match

1. Introduction

In the past decade, deep learning has dominated the machine learning landscape in data classification [1,2], predictive analysis [3], recommendation system [4], anomaly detection [5,6] and so on. Within deep learning, supervised classification methods have significantly improved the performance of deep learning in various classification tasks. However, it is still very difficult to obtain labels provided by professionals in many big data application scenario. In contrast, unsupervised classification methods have obvious advantages in dealing with unlabeled samples. Nevertheless, they greatly sacrificed the accuracy of the model because can not directly evaluate and optimize the performance of deep learning models by using the label information. Therefore, semi-supervised learning [7] caught the attention of researchers, which significantly improved the model performance of unsupervised learning by leveraging datasets with a small amount of label data. A more practical value lies in that semi-supervised learning methods can reduce the cost and time of manually marking data. Recently, many semi-supervised learning methods based on deep learning have been proposed [8–10], which can achieve quite good performance by leveraging the small fraction of labeled samples in the dataset.

However, semi-supervised learning faces two major challenges: *i*) how to transfer information obtained from limited labeled data to unlabeled data, and *ii*) how to learn as accurate information

as possible directly from a large amount of unlabeled data. To address these issues, semi-supervised learning uses three key loss terms (i.e., entropy minimization, generalization regularization, and consistency regularization) to better motivate the model to learn the corresponding downstream tasks. Among them, entropy minimization can encourage the model to confidently predict the output of unlabeled data, thereby improving the accuracy and robustness of the model. Generalization regularization constrains the model's parameters to avoid overfitting to the training data during the training process, thereby improving the model's generalization performance. According to the consistency assumption, data points that are close to each other often have consistency in the same label and structure [11]. Therefore, consistency regularization can improve the accuracy and generalization ability of the model by making the data consistent on the manifold.

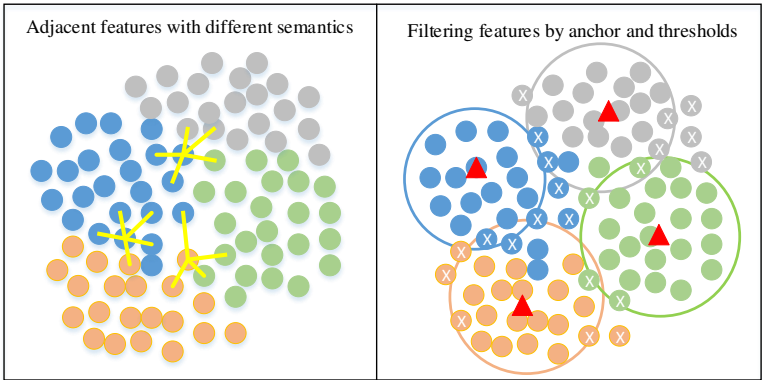
In addition, many studies have explored methods for solving the information propagation problem between different data [12], providing ideas for addressing the first major issue in semi-supervised learning. Among them, pseudo-labeling and consistency regularization methods provide ideas for solving the second major issue in semi-supervised learning. The pseudo-labeling method [13] uses the predictions of a classification model or a clustering algorithm as artificial labels to retrain the model. The consistency regularization method [9,10,14] forces the model to make the same prediction for the same sample under different transformations, learning from unlabeled data. However, these semi-supervised learning methods do not consider the possibility of introducing different levels of erroneous information during the training process, which can lead to low classification accuracy.

In the feature extraction stage, as plotted in Figure 1(a)(left) the model may have difficulty accurately differentiating between semantic differences at the classification boundary due to the ambiguity of feature representation boundaries. Without setting anchors for each class, the model may learn incorrect semantic information. However, by providing each class with an anchor, confidence thresholds can be set using the similarity between each sample and the anchor, allowing low-confidence features to be filtered out as shown in Figure 1(a)(right). When assigning pseudo-labels to unlabeled data, as plotted in Figure 1(b), assigning labels to samples with low predicted confidence may lead to confirmation bias [15], where the model overfits to incorrect labels and reduces its performance. Using a fixed threshold method cannot adapt to the dynamic changes of the dataset. On the contrary, a global dynamic threshold method dynamically adjusts the threshold based on the confidence distribution of unlabeled samples in the current iteration, thus avoiding this problem.

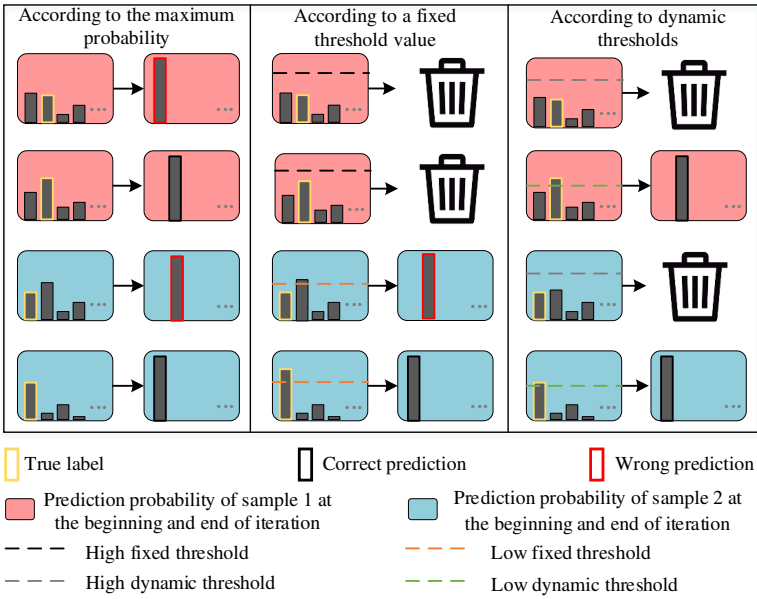
To address these issues, we propose ReliaMatch, a semi-supervised classification method that filters unreliable information based on a confidence threshold. ReliaMatch adopts a confidence threshold filtering strategy, which matches the similarity of labeled data and unlabeled data in feature space by setting anchor points, thus filtering out outliers and demarcation points with ambiguous semantics. Dynamic threshold is used to select reliable pseudo-labels, so as to eliminate the confirmation deviation of the model to pseudo-labels and improve classification performance. Additionally, ReliaMatch adopts the training mode of Curriculum Learning [16], which combines the screened samples and their pseudo-labels with labeled data, gradually increasing the difficulty of training data sets and participating in model training in a supervised way, thus further improving the classification performance. In summary, we make the following three main contributions:

- 1) We propose a semi-supervised classification method (Reliable Match), which addresses the issue of confirmation bias that arises from unlabeled data having different semantics and low prediction confidence near the classification boundary.
- 2) ReliaMatch employs a confidence threshold filtering strategy that matches the similarity of labeled and unlabeled data in feature space by setting anchor points, which filters out outliers and demarcation points with ambiguous semantics. To eliminate confirmation deviation of the model to pseudo labels and improve classification performance, ReliaMatch uses a dynamic threshold to select reliable pseudo-labels.

3) ReliaMatch employs Curriculum Learning training mode, which combines the screened samples and their pseudo-labels with labeled data and gradually increases the difficulty of the training dataset, thereby participating in model training in a supervised manner and further improving classification performance.



(a) The left figure shows hows the semantic inconsistency between adjacent samples, and the right figure illustrates how to use anchors to set confidence thresholds to filter features.



(b) Comparison of different strategies for pseudo-label generation. The maximum predicted probability strategy may cause confusion in the early stages but improves in the later stages. The fixed threshold strategy may lead to either insufficient or incorrect labeling. The dynamic threshold strategy, however, can adjust the threshold based on the confidence of unlabeled samples and achieve higher accuracy in pseudo-labeling.

Figure 1. Unreliable information that may be introduced in the model training process.

2. Related Work

2.1. Semi-supervised Classification

Semi-supervised learning (SSL) has been extensively studied in various fields, including image classification [17], object detection [18], and semantic segmentation [19]. SSL methods in image classification aim to reduce reliance on labeled data by leveraging unlabeled data. In SSL, labeled results are typically obtained through consistent regularization [20–22], pseudo-labeling [13], and entropy minimization [23]. Consistent regularization ensures that the model produces consistent predictions

for different transformations of the same image. Pseudo-labeling employs model confidence to assign labels and guide the training process, while entropy minimization encourages the model to produce highly confident predictions. These labeling strategies have been widely adopted in many SSL approaches.

2.2. Consistency regularization

Consistency regularization plays a crucial role in modern semi-supervised learning (SSL) algorithms. The core idea behind consistency regularization is that the same input sample should produce consistent outputs under different perturbations. Early works such as [9,24,25] proposed this concept, which was further developed in [8,14,22]. The fundamental form of consistency regularization in SSL is often achieved through a loss term. The equation below represents this basic form:

$$\|p_m(y|A(x);\theta) - p_m(y|A(x);\theta)\|_2^2, \quad (1)$$

where A refers to stochastic functions, resulting in different values for $A(x)$, while p_m represents the model's output probability. In [25], random data augmentation, dropout, and random maximum pooling are employed as A to ensure similarity among the predictions of neural networks. On the other hand, [10] adopts adversarial transformations for augmentation. Another related approach, presented in [9], extends the perturbations to different time periods, requiring the current prediction of a sample to be similar to the prediction set of the same sample in the past. These perturbations mainly arise from different network states and data augmentations.

In SSL, consistency regularization techniques aim to ensure that the same input sample produces consistent output predictions under different perturbations. Different approaches have been proposed to achieve this goal. For example, in [25], random data augmentation, dropout, and random maximum pooling are used to promote similarity among the predictions of neural networks. Adversarial transformations are employed for augmentation in [10]. Additionally, [9] extends the perturbations to different time periods, enforcing similarity between current and past predictions for the same sample. In [14], two networks with the same structure are utilized, and the consistency constraint is enforced by comparing the predicted distributions using KL divergence or cross-entropy functions. This approach is further developed in [26], where uncertainty weighting is applied to unlabeled samples, focusing on samples with lower uncertainty. Virtual adversarial training, proposed by Miyato et al. [10], introduces adversarial noise as interference into data samples, followed by unified regularization of the resulting predictions. Another recent idea by Luo et al. [27] suggests using a comparison loss as the regularization term, ensuring that predictions from the same (or different) categories are similar (or different). This extends the scope of consistency regularization to cover consistency between different samples and can be combined with other methods like [14] or [10] for improved performance. To address model memorization and sensitivity to adversarial data, Mixup, proposed by Zhang et al. [28], pairs examples and labels by training a convex combination of neural networks. Verma et al. [29] build on Mixup with interpolation consistency training, which encourages consistency between unlabeled samples and the interpolation prediction of a single sample. Moreover, in [22], consistent regularization is achieved through estimating low-entropy labels, generating data-augmented unlabeled samples, and utilizing Mixup to combine labeled and unlabeled samples.

2.3. Pseudo-labeling

Pseudo labels are artificial labels generated by the model itself, which are used to further train the model. Through the pseudo-labeling method, we can use both labeled samples and pseudo labeled samples as new training data to update the model, thus greatly improving the utilization rate of unlabeled samples.

Lee et al. [13] chose the class with the highest prediction probability of the model as the pseudo label, however, pseudo labels are only used in the fine-tuning stage, and the network needs to be

pre-trained. Mandal et al. [30] propose a new deep semi-supervised framework, which can seamlessly process marked and unlabeled data. The framework is trained by two parts in turn: firstly, the label prediction component is used to predict the label of the unlabeled part of the training data, and then the common representation of two patterns is learned for cross-modal retrieval. Caron et al. [31] proposed a deep clustering algorithm combining K-means clustering algorithm and convolutional neural network, which used the clustering results of K-means with unlabeled data as false labels to assist CNN in classification. Based on the extreme value theory, Cascante-Bonilla et al. [32] put forward the Curriculum Labeling (CL), which uses careful curriculum selection as pacing standard to strengthen the pseudo labeling. Hu *et al.* [33] design a new end-to-end Iterative Feature Clustering Graph Convolution Network (IFC-GCN) to enhance the standard GCN through the iterative feature clustering module, and design an EM-like framework to improve the network performance by alternately correcting false labels and the node characteristics.

3. Method

The core idea of ReliaMatch is to match the correlation between labeled and unlabeled data, by filtering reliable unlabeled data and generating pseudo-labels, which are then used as new training data in the supervised learning process of the model. The detailed process of ReliaMatch is shown in Figure 2.

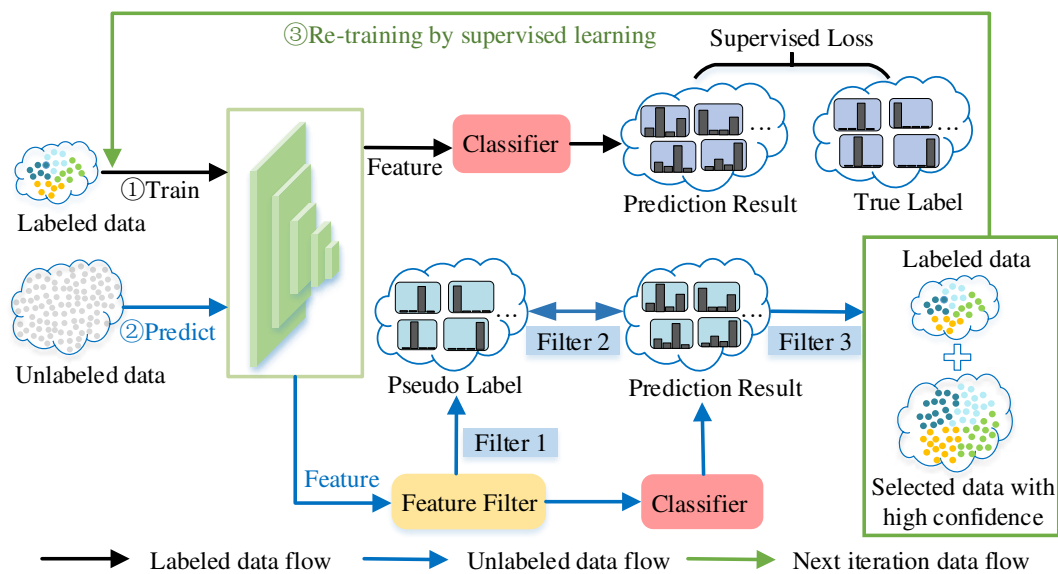


Figure 2. Illustration of the ReliaMatch framework.

This method uses confidence threshold to filter unreliable information, that is, feature vectors on the boundary of unlabeled data classification or outlier feature vectors and artificially marked false labels that may be incorrect. ReliaMatch adopts a self-training framework, that is, by iteratively learning the information in unlabeled datasets and labeled datasets, the performance of deep learning model is improved. In the training process, ReliaMatch uses the trained model to predict the unlabeled data, and adds the reliable data points in the prediction results and their pseudo-labels to the pseudo-label dataset. Then the pseudo-labeled dataset and the labeled dataset are merged to train the next round model. This process is repeated until a preset number of iterations or performance convergence is achieved.

For labeled samples, ReliaMatch trains them using the feature extraction model and the classification model. After training, feature extraction and classification prediction are performed on the labeled samples. At the same time, the average feature vector of each category of the labeled

samples is calculated and used as a feature anchor for filtering reliable features of the unlabeled samples.

For unlabeled samples, ReliaMatch first performs data processing by using data augmentation techniques to expand the unlabeled samples. Next, the enhanced unlabeled samples are input into the model for feature extraction. Then, the feature filter module calculates the similarity between the extracted features of the unlabeled samples and the feature anchor, and sets a feature similarity threshold to filter out the unlabeled samples with low similarity. This process is called filtering 1. The unlabeled samples selected by filtering 1 are labeled with the class label of the nearest feature anchor, and are used as pseudo-labels (hard labels). Next, ReliaMatch inputs these feature-filtered samples into classifier for classification prediction. For each unlabeled sample, it compares whether the class label of its maximum predicted probability (soft label) is consistent with the class label of the pseudo-label. If the classes are inconsistent, the unlabeled sample is filtered out. This process is called filtering 2. Finally, a dynamic threshold is set for the predicted probability to filter out unlabeled samples with maximum predicted probability below the threshold. This process is called filtering 3.

After three rounds of filtering, the remaining unlabeled samples are considered to be high-confidence reliable samples and are combined with their pseudo-labels to form a pseudo-labeled dataset. These pseudo-labeled samples are merged with labeled samples to form a new labeled dataset, which is used to train a new model. This process is iterated continuously to gradually increase the size of the labeled dataset and improve the performance of semi-supervised learning.

3.1. Problem Description

To describe the design process of the ReliaMatch model more accurately, we assume that in the t^{th} round of iteration, the training dataset X_N^t is used, which contains N^t samples including image data from different categories $X_N = \{x_i\}_{i=1}^{N^t}$. The training dataset is divided into a labeled dataset $X_L^t = \{x_{i,l}^t\}_{i=1}^{N_L^t}$ and an unlabeled dataset $X_U^t = \{x_{i,u}^t\}_{i=1}^{N_U^t}$. Assuming that f_θ^t is the convolutional neural network used for feature extraction in the t^{th} round of iteration. $Z_N^t = f_\theta^t(x_i^t)$ represents the feature vector set obtained from X_N^t after being processed by the convolutional neural network f_θ^t . Z_N^t consists of two parts, Z_L^t and Z_U^t , where $Z_L^t = f_\theta^t(x_{i,l}^t, y_{i,l}^t) = (z_{i,l}^t, y_{i,l}^t)_{i=1}^{N_L^t}$ is the labeled feature vector set in the t^{th} round of iteration, and $Z_U^t = f_\theta^t(x_{i,u}^t) = \{z_{i,u}^t\}_{i=1}^{N_U^t}$ is the unlabeled feature vector set in the t^{th} round of iteration. Let g_ω^t be the fully connected neural network (classifier) used by the model for classification prediction in the t^{th} round of iteration, with its output being the predicted probability of the sample in each category. Let $p_{i,j}^t$ denote the probability that the model predicts sample x_i^t as category j , and K is the number of categories. Then $P(x_i^t) = g_\omega^t(f_\theta^t(x_i^t)) = \{p_{i,j}^t\}_{j=1}^K$.

3.2. Feature Anchoring

Our first contribution is feature anchoring, which uses the features of the labeled data to calculate the average of the features of each category as an anchor. We think that nearby points are likely to have the same labels, and we also agree points on the same structure (usually called clusters or manifolds) may have the same label. Therefore, our method pay attention to the representation study of images, and we use the similarity of feature level to filter unreliable samples and assigns pseudo labels to each reliable sample. A schematic of filtering out poor feature of samples can be seen in Figure 3.

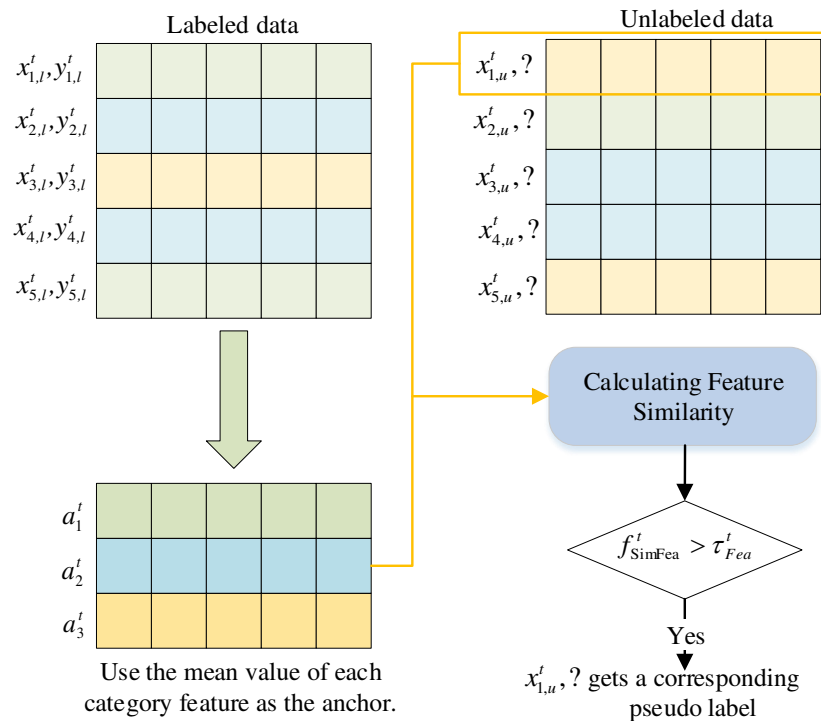


Figure 3. Schematic diagram of anchor generation and feature filtering. In the t^{th} iteration, five labeled samples and five unlabeled samples are selected, which come from four categories, and the samples with the same color indicate that they belong to the same category.

We first calculate the average feature vectors of each class j in the labeled feature dataset Z_L^t at the t^{th} iteration to generate K feature anchor points $A^t = \{a_j^t\}_{j=1}^K$. To do this, we use the following equation:

$$a_j^t = \frac{1}{|Z_{j,L}^t|} \sum_{(z_{i,l}^t, y_{i,l}^t) \in Z_{j,L}^t} z_{i,l}^t, \quad (2)$$

where $Z_{j,L}^t$ represents a subset of feature samples in the feature vector set Z_L^t where the label $y_{i,l}^t = j$ in the t^{th} round of iteration. $|Z_{j,L}^t|$ denotes the number of samples in the subset $Z_{j,L}^t$.

Next, we use cosine similarity to calculate the similarity between the extracted enhanced unlabeled samples' features and feature anchor points:

$$\begin{aligned} f_{SimFea}^t &= s(z_{i,u}^t, a_j^t) = \frac{z_{i,u}^t \cdot a_j^t}{|z_{i,u}^t| \cdot |a_j^t|} \\ &= \frac{\sum_{d=1}^D z_{i,u,d}^t a_{j,d}^t}{\sqrt{\sum_{d=1}^D (z_{i,u,d}^t)^2} \cdot \sqrt{\sum_{d=1}^D (a_{j,d}^t)^2}}, \end{aligned} \quad (3)$$

where $z_{i,u}^t$ represents the feature vector of the unlabeled data, a_j^t represents the anchor point, \cdot represents the dot product of the vector, $|\cdot|$ represents the norm of the vector, D is the dimension of the vector. A cosine similarity close to 1 indicates that the two vectors are very close in space, while a cosine similarity close to -1 indicates that the two vectors are almost opposite in space. A cosine similarity close to 0 indicates that there is no obvious correlation between the two vectors in space.

Next, the minimum similarity between labeled sample features and anchor points is used as the threshold for feature similarity, and a hyperparameter is used to dynamically adjust the threshold size of feature similarity:

$$\tau_{Fca}^t = \alpha_{Fca}^t \min(s(z_{i,l}^t, a_j^t)) = \alpha_{Fca}^t \min\left(\frac{z_{i,l}^t \cdot a_j^t}{|z_{i,l}^t| \cdot |a_j^t|}\right), \quad (4)$$

where $\alpha_{Fca}^t \in [0, 1]$ is a coefficient used to dynamically adjust the threshold of feature similarity, $z_{i,l}^t$ represents the feature vector of the labeled sample, a_j^t represents the feature anchor point, \cdot denotes the dot product of vectors, and $|\cdot|$ denotes the norm of vectors. The feature similarity threshold $\tau_{Fca}^t \in [-1, 1]$.

If the similarity f_{SimFca}^t between the feature of an unlabeled sample and the feature anchor point is greater than the feature similarity threshold τ_{Fca}^t , that is, $f_{SimFca}^t > \tau_{Fca}^t$, then the pseudo-label of the current sample $x_{i,u}^t$ is considered to be the label j of the feature anchor point that is most similar to its feature vector.

In the feature filtering module, the labeled dataset remains unchanged. In the t -th iteration, pseudo-labels of unlabeled samples can be obtained after feature filtering. The feature-filtering module dataset consisting of the filtered unlabeled samples and their pseudo-labels is denoted as $X_{U1}^t = \{(x_{i,u1}^t, y_{i,u1}^t)\}_{i=1}^{N_{U1}^t}$, where $y_{i,u1}^t$ is the pseudo-label of $x_{i,u1}^t$, and N_{U1}^t is the number of samples in the dataset.

3.3. Dynamic Allocation Pseudo Labels

In the pseudo-labeling strategy, threshold selection is an important issue. Traditional semi-supervised classification methods usually use a fixed threshold to predict the class of unlabeled samples and assign pseudo-labels to high-confidence samples above this threshold for training. However, this method may not adapt well to changes in the data distribution, and may either filter out useful samples excessively or add unreliable pseudo-labels, leading to classification errors due to samples of different classification difficulties.

To address these issues, the pseudo-label filtering module in ReliaMatch adopts a global dynamic threshold to filter reliable samples. Specifically, in each iteration, the module sets the confidence threshold for predicting probability based on the average confidence of unlabeled samples for the entire dataset. If the average confidence of unlabeled samples is high, it indicates that the algorithm has good classification performance for unlabeled samples, and the threshold can be increased. If the change in labels between two iterations is significant, it indicates that the algorithm has not yet converged and the threshold should be appropriately lowered to ensure the accuracy of the model.

For each unlabeled sample $x_{i,u1}^t$, its predicted probabilities $P(x_{i,u1}^t) = \{p_{i,u1,j}^t\}_{j=1}^K$ can be converted into a hard label by creating a vector of length K , denoted as $O_i^t = \{o_{i,j}^t\}_{j=1}^K$, as follows:

$$o_{i,j}^t = \begin{cases} 1, & \text{if } j = \arg \max(p_{i,u1,j}^t) \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

where $o_{i,j}^t$ indicates the label assigned to sample $x_{i,u1}^t$ for the j^{th} class. Specifically, the class with the highest predicted probability is assigned a label of 1 and the others are assigned a label of 0.

Next, the average confidence score p_u^t can be obtained by computing the average of the maximum predicted probability values for all unlabeled samples, i.e.,

$$p_u^t = \frac{1}{|X_{U1}^t|} \sum_{x_{i,u1} \in X_{U1}^t} \max(P(x_{i,u1}^t)), \quad (6)$$

where X_{U1} denotes the unlabeled dataset after feature filtering, and $P(x_{i,u1}^t)$ represents the predicted probabilities of sample $x_{i,u1}$ in the t^{th} training round.

ReliaMatch uses the value of p_u^t as the confidence level of unlabeled data to adjust the credibility threshold of pseudo labels to ensure the quality of pseudo labels. Therefore, the global dynamic threshold τ_{pre}^t in round t can be represented as:

$$\tau_{pre}^t = \begin{cases} \max\left(\frac{1}{1+e^{-p_u^t}}, \tau_{pre}^0\right), & t = 1 \\ \max\left(\frac{1}{1+e^{-p_u^t}}, \frac{p_u^t \cdot \|y_{i,u1}^t - y_{i,u1}^{t-1}\|_F}{\|p_u^t(x_{i,u1}^t) - p_u^{t-1}(x_{i,u1}^{t-1})\|_F + \epsilon}\right), & t > 1, \end{cases} \quad (7)$$

where τ_{pre}^0 is the initial threshold. ϵ is a very small constant to avoid division by zero. $\|\cdot\|_F$ denotes matrix norm, and p_u^t represents the average confidence level in round t . $\|y_{i,u1}^t - y_{i,u1}^{t-1}\|_F$ represents the size of the difference between the pseudo labels of the unlabeled sample i in round t and those in round $t - 1$. $\|p_u^t(x_{i,u1}^t) - p_u^{t-1}(x_{i,u1}^{t-1})\|_F$ represents the difference between the confidence level p_u^t of the unlabeled sample i in round t and that in round $t - 1$.

To select the most reliable pseudo labels, ReliaMatch further filters the unlabeled data in X_{U1}^t . The filtering criteria are as follows:

- Only select samples whose maximum predicted probability is greater than the predicted probability threshold τ_{pre} :

$$\max(P(x_{i,u1}^t)) > \tau_{pre}. \quad (8)$$

- Only select samples whose predicted category is consistent with the pseudo label in the pseudo label filtering module:

$$O_i^t = y_{i,u1}^t. \quad (9)$$

After screening, only samples that meet the above requirements can be added to the labeled dataset and used for supervised network training. Therefore, the resulting pseudo-labeled dataset is denoted as $X_{U2}^t = (x_{i,u2}^t, y_{i,u2}^t)_{i=1}^{N_{U2}^t}$, where $y_{i,u2}^t$ represents the pseudo-label of $x_{i,u2}^t$, and N_{U2}^t represents the number of samples in this dataset. The formula is as follows:

$$X_{U2}^t = \{(x_{i,u2}, y_{i,u2})\}_{i=1}^{N_{U2}^t} | \max(P(x_{i,u1}^t)) > \tau_{pre} \cap O_i^t = y_{i,u1}^t. \quad (10)$$

3.4. Loss

In ReliaMatch, the unlabeled dataset X_{U2}^t , after undergoing feature filtering and pseudo-label filtering, is transferred to the labeled dataset for supervised training, forming a new labeled dataset X_L^{t+1} and an updated unlabeled dataset X_U^{t+1} , given by the following formulas:

$$X_L^{t+1} = \{x_{i,l}^{t+1}, y_{i,l}^{t+1}\}_{i=1}^{N_L^t + N_{U2}^t} := X_L^t + X_{U2}^t, \quad (11)$$

$$X_U^{t+1} = \{x_{i,u}^{t+1}, y_{i,u}^{t+1}\}_{i=1}^{N_U^t - N_{U2}^t} := X_U^t - X_{U2}^t. \quad (12)$$

Based on this, the model parameters are updated using the supervised loss \mathcal{L}_s^t . Specifically, for a sample $x_{i,l}^t$ in the labeled dataset with its label $y_{i,l}^t$, we have:

$$\mathcal{L}_s^t = \mathcal{L}_{CE}(g_\omega^t(f_\theta^t(x_{i,l}^t)), y_{i,l}^t), \quad (13)$$

where $x_{i,l}$ is a sample from the labeled dataset in the t^{th} iteration, and $y_{i,l}$ is the label of this sample. \mathcal{L}_{CE} represents the cross-entropy loss function.

4. Experiments

4.1. Datasets

CIFAR-10: A dataset containing 60K images, with the shape of 32×32 , is evenly distributed in ten categories. There are 50K images in the training set and 10K images in the test set. Our validation set size is 5000 for CIFAR-10.

SVHN: A dataset containing 99289 images, with the shape of 32×32 , is evenly distributed in ten categories. SVHN consists of ten categories. The training contents contains 73257 images, and the test set contains 26032 images. Our validation set size is 5000 for SVHN.

4.2. Model Details

ReliaMatch uses CNN-13 [34] and WideResNet-28 [35] for classification on CIFAR-10 and SVHN datasets. In order to make fair comparisons with other methods, the same parameters as in [32] are used in this paper. The network optimization is performed using stochastic gradient descent and Nesterov momentum algorithm, combined with weight decay regularization to reduce overfitting. The momentum factor is set to 0.9 and the initial learning rate is 0.1. To further improve optimization, cosine annealing [36] is employed to update the model parameters. By default, the initial hyperparameters α_{Fea}^t is set to 1 and τ_{Pre}^0 is set to 0.95.

4.3. Experimental Results and Analysis

In this section, we compare ReliaMatch with other common semi-supervised learning methods, including Pseudo-Label [13], LP-MT [12], PL-CB [15], Curriculum Labeling [32] based on pseudo-labeling, π Model [9], Temporal Ensembling [9], Mean Teacher [14], VAT [10], Ladder Net [8], ICT [37] based on consistency regularization, and MixMatch [22] based on strong mixup.

We compared ReliaMatch with other commonly used semi-supervised learning methods, including Pseudo-Label [13], LP-MT [12], PL-CB [15], Curriculum Labeling [32], π Model [9] and Temporal Ensembling [9], Mean Teacher [14], VAT [10], Ladder Net [8] and ICT [37], and MixMatch [22], based on the WideResNet-28 and CNN-13 architectures for semi-supervised classification on the CIFAR10 and SVHN datasets.

From the results shown in Table 1 and Table 2, it can be seen that, compared to other semi-supervised learning methods, ReliaMatch considers issues such as feature and label noise, filters out semantically ambiguous features and unreliable pseudo-labels, and thus performs better. Specifically, the ReliaMatch method uses the same WideResNet-28 network as advanced semi-supervised classification methods proposed by previous researchers. On the CIFAR-10 dataset, this method used 4000 labeled data for testing, and achieved a test error rate of only 5.86%. In comparison, the test error rate of the Pseudo-Label method was 17.78%, Curriculum Labeling method was 8.92%, and PL-CB method was 6.28%. On the SVHN dataset, the ReliaMatch method used only 1000 labeled data for testing, achieving a test error rate of 4.04%, significantly better than the test error rates of Pseudo-Label method (7.62%) and Curriculum Labeling method (5.65%).

Table 1. Comparison of Test Error Rate(%) of WideResNet-28 with Different Semi-supervised Methods.

| Method | CIFAR-10($N_L = 4000$) | SVHN($N_L = 1000$) |
|----------------------------------|-----------------------------------|-----------------------------------|
| PL [†] | 17.78 ± 0.57 | 7.62 ± 0.29 |
| Curriculum Labeling [†] | 8.92 ± 0.03 | 5.65 ± 0.11 |
| PL-CB [†] | 6.28 ± 0.30 | - |
| π Model [†] | 16.37 ± 0.63 | 7.19 ± 0.27 |
| Mean Teacher | 10.36 ± 0.28 | 5.65 ± 0.47 |
| VAT [†] | 13.86 ± 0.27 | 5.63 ± 0.20 |
| VAT+EntMin [†] | 13.13 ± 0.39 | 5.35 ± 0.19 |
| ICT [†] | 7.66 ± 0.17 | 3.53 ± 0.07 |
| MixMatch [†] | 6.24 ± 0.06 | 3.27 ± 0.31 |
| ReliaMatch[*] | 5.86 ± 0.12 | 4.04 ± 0.08 |

N_L represents the number of labeled samples in the training set, [†] indicates that the result has been reported in the literature [32], and ^{*} represents the average of 5 runs of the proposed method in this paper.

Table 2. Comparison of Test Error Rate (%) of Different Semi-supervised Methods Using CNN-13.

| Method | CIFAR-10($N_L = 4000$) | SVHN($N_L = 1000$) |
|----------------------------------|-----------------------------------|-----------------------------------|
| LP-MT [†] | 10.16 ± 0.28 | - |
| Curriculum Labeling [†] | 9.81 ± 0.22 | 4.75 ± 0.28 |
| Ladder Net [†] | 12.16 ± 0.31 | - |
| Temporal Ensembling [†] | 12.16 ± 0.24 | 4.42 ± 0.16 |
| ReliaMatch[*] | 7.42 ± 0.05 | 7.13 ± 0.28 |

N_L represents the number of labeled samples in the training set, [†] indicates that the result has been reported in the literature [32], and ^{*} represents the average of 5 runs of the proposed method in this paper.

In addition, the ReliaMatch method uses the same CNN-13 network as advanced semi-supervised classification methods proposed by previous researchers. On the CIFAR-10 dataset, using 4000 labeled data for testing, the test error rate of ReliaMatch method was 7.42%, which is lower than the test error rates of Pseudo-Label methods (LP-MT, Curriculum Labeling) and consistency regularization methods (Ladder Net, Temporal Ensembling).

One common way to evaluate semi-supervised classification algorithms is by varying the size of the labeled dataset. By reducing the number of available labeled samples, it is possible to better simulate real-world scenarios. On the CIFAR-10 and SVHN datasets, we tested the error rates of the ReliaMatch method using the WideResNet-28 network under different numbers of labeled samples. We used datasets with 500, 1000, 2000, and 4000 labeled samples, respectively, and only changed the number of labeled samples during each training while keeping other hyperparameters the same as when using 4000 labeled samples.

The experimental results show that the classification performance of the ReliaMatch method does not significantly degrade under different numbers of labeled samples. This indicates that the ReliaMatch method has good robustness and can provide stable performance even with very limited labeled data, which is crucial for practical applications since in many cases, only very limited labeled data can be obtained. Therefore, these results suggest that the ReliaMatch method is an effective semi-supervised classification algorithm that can be useful in practical applications.

In addition, we also investigated the effectiveness of the feature filtering module and pseudo-label filtering module in the ReliaMatch method. As shown in Table 3, we separately or collectively removed these two modules and evaluated their impact on the method performance by applying data augmentation during training. The experiments were conducted on the CIFAR-10 dataset with 4000 labeled samples, and the WideResNet-28 was used as the backbone network.

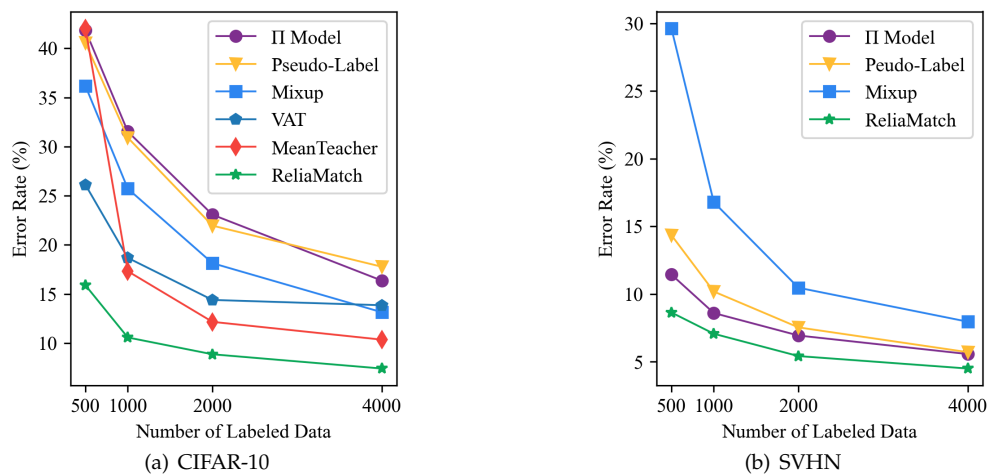


Figure 4. Comparison of test error rates of different methods using different amounts of labeled data.

Table 3. Explore the influence of important modules of ReliaMatch on classification results.

| Method | Test Error Rate(%) |
|--------------------------------------------------|--------------------|
| w/o Feature Filtering | 6.74 |
| w/o Pseudo-label Filtering | 9.12 |
| w/o Feature Filtering and Pseudo-label Filtering | 16.9 |
| ReliaMatch(benchmark) | 5.86 |

After analyzing the Table 3, we have reached the following conclusions: the pseudo label filtering module has a significant impact on the performance of the ReliaMatch method, as the removal of this module leads to an increase in model error rate from 5.86 to 9.12. This indicates that the pseudo label filtering module can remove low-confidence pseudo labels, thereby reducing noise and improving model performance. In contrast, the effect of the feature filtering module is not as significant as that of the pseudo label filtering module, but it still contributes to improving model performance. Its removal results in a model error rate increase from 5.86 to 6.74, indicating that the feature filtering module can select high-quality features and thus improve model performance. Additionally, removing both feature and pseudo label filtering modules causes a sharp increase in model error rate, from 5.86 to 16.9. This demonstrates that both feature and pseudo label filtering modules make important contributions to the performance of the ReliaMatch method, and both are necessary.

Therefore, both feature and pseudo label filtering modules play important roles in the ReliaMatch method. The feature filtering module can select high-quality features, while the pseudo label filtering module can remove low-confidence pseudo labels. Their combined effect can improve model performance.

We demonstrated the training process of ReliaMatch using the CNN-13 network on the CIFAR-10 and SVHN datasets. ReliaMatch adopts a curriculum learning training approach. The method combines the selected samples and their reliable pseudo-labels with the labeled data, gradually increasing the difficulty of the training data to participate in model training in a supervised manner. As shown in Figures 5 and 6, during the training process, ReliaMatch reinitializes the model training after each label transfer to mitigate the problem of confirmation bias caused by pseudo-labels.

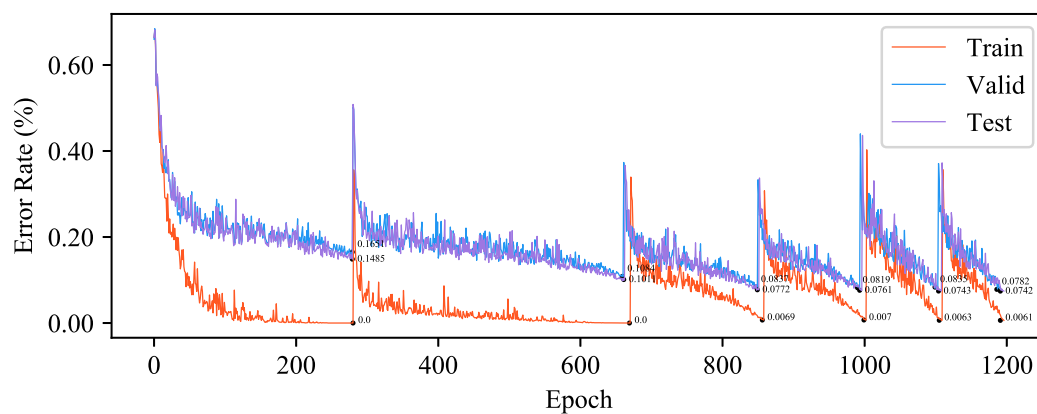


Figure 5. Training process of ReliaMatch on CIFAR-10 dataset.

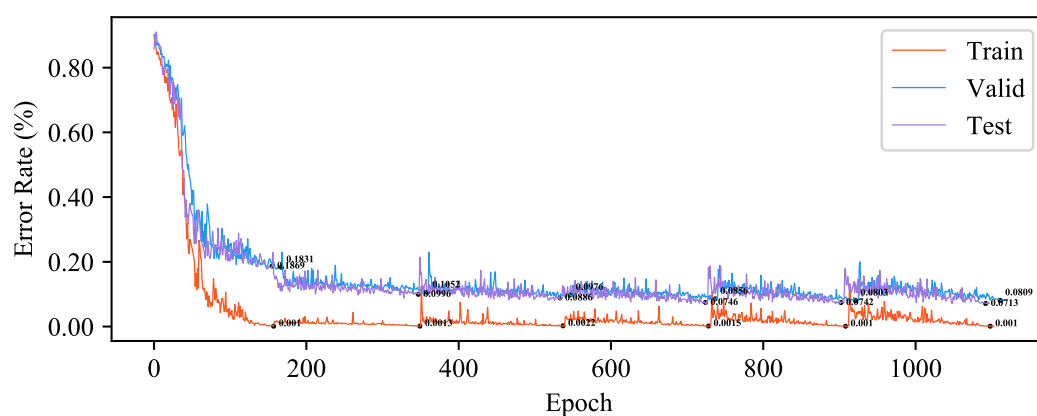


Figure 6. Training process of ReliaMatch on SVHN dataset.

5. Conclusions

In this paper, we proposed a reliable semi-supervised deep learning classification algorithm, i.e., ReliaMatch. The algorithm integrates course label, feature filter module and pseudo-label filter module, aiming at improving classification accuracy and algorithm reliability, focusing on key features, making better use of unlabeled data and avoiding the confirmation deviation of pseudo-labels. The course label improves the classification accuracy and the reliability of the algorithm. The feature filtering module eliminates unnecessary features, which makes the algorithm pay more attention to key features. The false label filtering module eliminates the confirmation deviation of false labels and filters out unreliable features and labels with low confidence, thus making the algorithm more stable and reliable. The experimental results show that ReliaMatch achieves the most advanced classification results on multiple data sets under the control of the confidence threshold.

Author Contributions: Conceptualization, T.J. and W.C.; methodology, T.J.; validation, L.C. and W.C.; data curation, W.C.; writing—original draft preparation, T.J. and W.C.; writing—review and editing, T.J., W.C., W.M. and P.Q.; visualization, L.C. and W.C.; supervision, T.J., W.M. and P.Q.; funding acquisition, T.J. and P.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China No.62171334 and Fundamental Research Funds for the Central Universities No. ZYTS23162.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: “The CIFAR-10 dataset” is available at <https://www.cs.toronto.edu/~kriz/cifar.html> and “The Street View House Numbers (SVHN) Dataset” is available at <http://ufldl.stanford.edu/housenumbers/>.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lee, H.; Kwon, H. Going deeper with contextual CNN for hyperspectral image classification. *IEEE Trans. Image Process.* **2017**, *26*, 4843–4855.
2. Qi, P.; Zhou, X.; Ding, Y.; Zhang, Z.; Zheng, S.; Li, Z. FedBKD: Heterogenous federated learning via bidirectional knowledge distillation for modulation classification in IoT-edge system. *IEEE Journal of Selected Topics in Signal Processing* **2023**, *17*, 189–204.
3. Wang, Y.; Long, M.; Wang, J.; Gao, Z.; Yu, P.S. PredRNN: recurrent neural networks for predictive learning using spatiotemporal LSTMs. *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*; Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H.M.; Fergus, R.; Vishwanathan, S.V.N.; Garnett, R., Eds., 2017, pp. 879–888.
4. Huang, W.; Wu, Z.; Liang, C.; Mitra, P.; Giles, C.L. A neural probabilistic model for context based citation recommendation. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*; Bonet, B.; Koenig, S., Eds. AAAI Press, 2015, pp. 2404–2410.
5. Ergen, T.; Kozat, S.S. Unsupervised anomaly detection with LSTM neural networks. *IEEE transactions on neural networks and learning systems* **2019**, *31*, 3127–3141.
6. Qi, P.; Jiang, T.; Wang, L.; Yuan, X.; Li, Z. Detection tolerant black-Box adversarial attack against automatic Modulation Classification With Deep Learning. *IEEE Transaction on Reliability* **2022**, *71*, 674–686.
7. Chapelle, O.; Scholkopf, B.; Zien, A. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks* **2009**, *20*, 542–542.
8. Rasmus, A.; Berglund, M.; Honkala, M.; Valpola, H.; Raiko, T. Semi-supervised learning with ladder networks. *Proceedings of the Advances in Neural Information Processing Systems; Cortes, C.; Lawrence, N.D.; Lee, D.D.; Sugiyama, M.; Garnett, R., Eds., 2015*, pp. 3546–3554.
9. Laine, S.; Aila, T. Temporal ensembling for semi-supervised learning. *Proceedings of the 5th International Conference on Learning Representations*, 2017.
10. Miyato, T.; Maeda, S.; Koyama, M.; Ishii, S. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2019**, *41*, 1979–1993.
11. Zhou, D.; Bousquet, O.; Lal, T.N.; Weston, J.; Schölkopf, B. Learning with local and global consistency. *Proceedings of the Advances in Neural Information Processing Systems*, 2004, pp. 321–328.
12. Iscen, A.; Tolias, G.; Avrithis, Y.; Chum, O. Label propagation for deep semi-supervised learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5070–5079.
13. Lee, D.H.; others. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. *Proceedings of the Workshop on Challenges in Representation Learning*, 2013, p. 896.
14. Tarvainen, A.; Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Proceedings of the 5th International Conference on Learning Representations*, 2017.
15. Arazo, E.; Ortego, D.; Albert, P.; O'Connor, N.E.; McGuinness, K. Pseudo-labeling and confirmation bias in deep semi-supervised learning. *Proceedings of the 2020 International Joint Conference on Neural Networks*, 2020, pp. 1–8.
16. Bengio, Y.; Louradour, J.; Collobert, R.; Weston, J. Curriculum learning. *Proceedings of the 26th Annual International Conference on Machine Learning; Danyluk, A.P.; Bottou, L.; Littman, M.L., Eds., 2009*, pp. 41–48.
17. Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, 20-25 June 2009, Miami, Florida, USA. IEEE Computer Society, 2009, pp. 248–255.
18. Lin, T.; Maire, M.; Belongie, S.J.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: common objects in context. *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*; Fleet, D.J.; Pajdla, T.; Schiele, B.; Tuytelaars, T., Eds. Springer, 2014, Vol. 8693, *Lecture Notes in Computer Science*, pp. 740–755.

19. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The cityscapes dataset for semantic urban scene understanding. 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. IEEE Computer Society, 2016, pp. 3213–3223.
20. Wang, K.; Yang, C.; Betke, M. Consistency regularization with high-dimensional non-adversarial source-guided perturbation for unsupervised domain adaptation in segmentation. Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021. AAAI Press, 2021, pp. 10138–10146.
21. Abuduweili, A.; Li, X.; Shi, H.; Xu, C.; Dou, D. Adaptive consistency regularization for semi-supervised transfer learning. IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021. Computer Vision Foundation / IEEE, 2021, pp. 6923–6932.
22. Berthelot, D.; Carlini, N.; Goodfellow, I.; Papernot, N.; Oliver, A.; Raffel, C. Mixmatch: A holistic approach to semi-supervised learning. Proceedings of the Advances in Neural Information Processing Systems, 2019, pp. 5050–5060.
23. Grandvalet, Y.; Bengio, Y. Semi-supervised learning by entropy minimization. Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada], 2004, pp. 529–536.
24. Bachman, P.; Alsharif, O.; Precup, D. Learning with pseudo-ensembles. Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada; Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N.D.; Weinberger, K.Q., Eds., 2014, pp. 3365–3373.
25. Sajjadi, M.; Javanmardi, M.; Tasdizen, T. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain; Lee, D.D.; Sugiyama, M.; von Luxburg, U.; Guyon, I.; Garnett, R., Eds., 2016, pp. 1163–1171.
26. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019.
27. Luo, Y.; Zhu, J.; Li, M.; Ren, Y.; Zhang, B. Smooth neighbors on teacher graphs for semi-supervised learning. 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. Computer Vision Foundation / IEEE Computer Society, 2018, pp. 8896–8905.
28. Zhang, H.; Cissé, M.; Dauphin, Y.N.; Lopez-Paz, D. Mixup: Beyond empirical risk minimization. Proceedings of the 6th International Conference on Learning Representations, 2018.
29. Verma, V.; Lamb, A.; Kannala, J.; Bengio, Y.; Lopez-Paz, D. Interpolation consistency training for semi-supervised learning. Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019; Kraus, S., Ed. ijcai.org, 2019, pp. 3635–3641.
30. Mandal, D.; Rao, P.; Biswas, S. Semi-supervised cross-Modal retrieval with label prediction. *IEEE Trans. Multim.* **2020**, *22*, 2345–2353.
31. Caron, M.; Bojanowski, P.; Joulin, A.; Douze, M. Deep clustering for unsupervised learning of visual features. Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIV; Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y., Eds. Springer, 2018, Vol. 11218, *Lecture Notes in Computer Science*, pp. 139–156.
32. Cascante-Bonilla, P.; Tan, F.; Qi, Y.; Ordonez, V. Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. Proceedings of the 35th AAAI Conference on Artificial Intelligence, 2021, pp. 6912–6920.
33. Hu, Z.; Kou, G.; Zhang, H.; Li, N.; Yang, K.; Liu, L. Rectifying pseudo labels: iterative feature clustering for graph representation Learning. CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021; Demartini, G.; Zuccon, G.; Culpepper, J.S.; Huang, Z.; Tong, H., Eds. ACM, 2021, pp. 720–729.
34. Springenberg, J.T.; Dosovitskiy, A.; Brox, T.; Riedmiller, M.A. Striving for simplicity: The all convolutional net. Proceedings of the 3rd International Conference on Learning Representations; Bengio, Y.; LeCun, Y., Eds., 2015.

35. Zagoruyko, S.; Komodakis, N. Wide residual networks. Proceedings of the British Machine Vision Conference 2016, 2016.
36. Loshchilov, I.; Hutter, F. SGDR: Stochastic gradient descent with warm restarts. Proceedings of the 5th International Conference on Learning Representations, 2017.
37. Verma, V.; Kawaguchi, K.; Lamb, A.; Kannala, J.; Solin, A.; Bengio, Y.; Lopez-Paz, D. Interpolation consistency training for semi-supervised learning. *Neural Networks* **2022**, *145*, 90–106.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.