Rethinking Safe Semi-supervised Learning: Transferring the Open-set Problem to A Close-set One

Qiankun Ma*[1], Jiyao Gao*[1], Bo Zhan[1], Yunpeng Guo[1], Jiliu Zhou[1] and Yan Wang†[1]

[1]School of Computer Science, Sichuan University, Chengdu, China

Abstract

Conventional semi-supervised learning (SSL) lies in the close-set assumption that the labeled and unlabeled sets contain data with the same seen classes, called in-distribution (ID) data. In contrast, safe SSL investigates a more challenging open-set problem where unlabeled set may involve some out-of-distribution (OOD) data with unseen classes, which could harm the performance of SSL. When we are experimenting with the mainstream safe SSL methods, we have a surprising finding that all OOD data show a clear tendency to gather in the feature space. This inspires us to solve the safe SSL problem from a fresh perspective. Specifically, for a classification task with K seen classes, we utilize a prototype network not only to generate K prototypes of all seen classes, but also explicitly model an additional prototype for the OOD data, transferring the K-way classification on the open-set to the (K+1)-way on the close-set. In this way, the typical SSL techniques (e.g., consistency regularization and pseudo labeling) can be applied to tackle the safe SSL problem without additional consideration of OOD data processing like other safe SSL methods do. Particularly, considering the possible low-confidence pseudo labels, we further propose an iterative negative learning (INL) paradigm to enforce the network learning knowledge from complementary labels on wider classes, improving the network's classification performance. Extensive experiments on four benchmark datasets show that our approach remarkably lifts the performance on safe SSL and outperforms the state-of-the-art methods.

1. Introduction

In the past decade, the development of deep learning brings prosperity to the field of computer vision, such as image classification, attributed to the growing scale of labeled data [18, 9, 23]. However, it is time-consuming and expensive to collect large amounts of labeled data. Numerous methods resort to semi-supervised learning (SSL) to relieve this restriction [33, 19, 32]. By exploring the massive valuable information from the abundant easy-to-acquire unlabeled data, SSL methods can effectively narrow the performance gap towards the fully-supervised models.

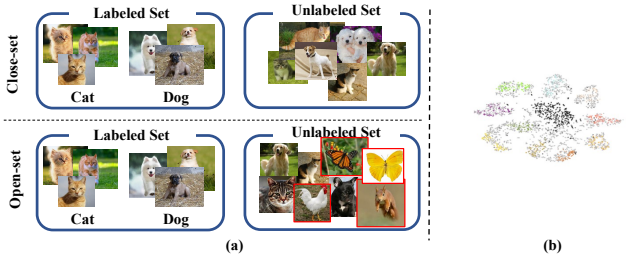

Figure 1. (a) An example of close-set and open-set. In open-set, the unlabeled set contains classes not seen in the labeled set (indicated by red boxes); (b) t-SNE [34] visualization of feature distributions of ID (colored) data and OOD (black) data from CIFAR10 [17], Images of the same category are shown in the same color.

Although the SSL methods are proven to be effective in improving the classification performance in case of sparse labels, most of them follow a close-set assumption by default, that is, the labeled data and the unlabeled data share the same label space. See Figure 1(a) for example, both labeled data and unlabeled data are built on the classes of cat and dog, which are also called in-distribution (ID) data. Yet, in many real scenarios, such a simple and crude assumption often breaks down since the unlabeled data could be collected in the wild [40]. In this case, there will be many samples with unknown classes (or unseen classes) appearing in the unlabeled dataset, which we call out-of-distribution (OOD) data, resulting in an open-set problem. As shown in Figure 1, in the open-set, besides the cat and dog classes in the labeled set (or seen classes), there are also some unseen classes, e.g., butterfly, hen, rabbit. Without reliable labels for these unseen classes, the OOD data may lead to a sig-

*Equal contribution.
†Corresponding author: Yan Wang (wangyanscu@hotmail.com).

nificant degradation of model performance [8, 4, 25], and even pose a significant safety risk to the accurate prediction since the model is forced to predict the unseen class as a seen one. This severely hinders the SSL methods from being deployed in real-world applications.

To handle this problem, a series of enhanced SSL methods have been emerged to improve the classification performance and guarantee the prediction safety in a more realistic scenario, namely safe semi-supervised learning (safe SSL) methods [8, 4, 10, 11, 41]. The *safe* here means that the model using extra unlabeled data will not be worse than a simple supervised one. Most current safe SSL methods adhere a two-step scheme: 1) identifying the OOD data, and 2) dealing with the OOD data. In stage one, these methods tend to identify the OOD data by classifying the OOD data and the ID data into two separate broad classes [8, 4, 10, 41]. After the OOD data is detected, some methods treat them as hazards and just discard them in the second stage [4, 41]. In contrast, He *et al*. [10] argued that the valuable information contained in the OOD data can be used to further improve the identification of OOD data. To this end, they endeavored to calibrate the seen-class probability distribution into a uniform distribution, thus suppressing the over-confidence problem to unseen class and eliminating the risk of hard OOD data being recognized as ID data.

Despite the effectiveness of the two-step pipeline, utilizing it in reality might still be cumbersome and requires consideration of some additional caveats. For instance, in stage one, clustering all ID data into one class is a coarse-grained binary classification task, which may conflict with the final fine-grained multi-class classification task in terms of feature space learning [14]. Furthermore, in stage two, it is not easy to come up with a feasible way to utilize OOD data to enhance model performance, either. When we are experimenting with these safe SSL methods, an interesting phenomenon strikes us as a surprise. We find that most safe SSL methods [8, 10, 14] show an intrinsic ability to gather the OOD features in the first stage like Figure 1(b), especially for [14] employing the self-supervised learning manner. This involkes us to ask the following question: Why the OOD data which belong to different classes are gathered in the feature space?

We try to answer this question by inspecting the Class Activation Map (CAM) [44] for ID and OOD images. As observed in Figure 2, the model is mainly concerned with the high-level information of the objects (*e.g*. the body of cats and birds) for ID data. For OOD data, on the other hand, due to the lack of corresponding categories of objects, the model can only pay more attention to some common category-agnostic low-level information, such as color, edge, and corner. Therefore, although the OOD data come from different classes, their features still present a gathering tendency in feature space.

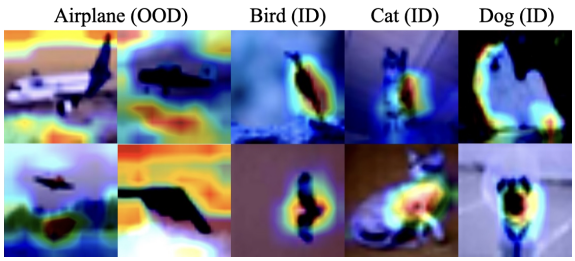

Figure 2. CAM visualization of the results. For the ID data (Bird, Cat, Dog), the features are more class-discriminative.

This inspires us to solve the safe SSL problem from a new viewpoint, that is we can treat all the OOD data as an individual class peer to other seen classes during training. In this way, we can transfer a K-way classification task on the open-set to a simple (K+1)-way classification task on the close-set. Then we can tackle the complex safe SSL task in the same way as the trivial SSL task using all the methods that are applicable on the close-set assumption, greatly streamlining the safe SSL problem.

Following the above inspiration, in this paper, we propose a prototype-based safe SSL framework for image classification from a fresh perspective. Specifically, our framework includes two stages, including the first stage of unseen-class prototype generation and the second stage of semi-supervised image classification. Assuming that there are K seen classes, the first stage starts by training a prototype network [31] on the labeled ID data so that it can produce K prototypes of seen classes. Based on the prototypes, we can distinguish the feature distribution of all the OOD data according to the distance of the features from the prototypes, thereby explicitly modeling an additional prototype of all unseen classes. These K+1 prototypes allow the framework to make predictions for both the K fine-grained seen classes and the remaining unseen class simultaneously, avoiding the conflict with coarse-grained classification. By casting the K-way classification to the (K+1)-way classification, we can directly use any common and effective SSL technique to carry out the semi-supervised classification task in the second stage.

A widely accepted solution for SSL is to filter high confidence pseudo labels for unlabeled data. Yet, removing all the unreliable predictions could lead to insufficient and categorically imbalanced training [37]. To make full use of unlabeled data, negative learning (NL) [37, 38, 16] is proposed to provide complementary labels which indicate the class to which the current sample is least likely to belong, namely negative class. However, the conventional NL may become less powerful when solving a classification task with a large number of classes. The reasons can be attributed to the following aspects: 1) Current NL methods don't ensure the complementary label of the same sample keep unchanged during the training process. Wavering labels are unfavor-

able for stable training; 2) When there are excessive classes, current NL methods can only determine the complementary labels on one negative class at a time, which can solely supplement limited knowledge to the network and leave massive information on other classes unexplored. To alleviate these limitations, we propose an iterative negative learning (INL) paradigm in the second stage. Unlike the previous NL methods [16, 37], we harness a memory bank to preserve the complementary labels of unlabeled data at each training iteration. As training goes, the complementary labels will explore new negative classes iteratively based on the saved historical complementary labels. By this, the INL paradigm enables to fully unearth the knowledge of all unlabeled data in the whole feature space, helping the model classify the samples more confidently. We summarize our contributions as follows:

- We rethink the safe SSL problem from a fresh perspective and propose a prototype-based safe SSL framework to explicitly model the OOD data as a novel class peer to ID classes. In this manner, we transform the safe SSL from an open-set problem to a close-set one.
- We raise an INL paradigm to enhance the feature learning capability of our framework with regard to the unreliable predictions in SSL. By employing a memory bank to progressively update the complementary label until it covers most negative classes, our model can excavate the knowledge of the unlabeled data in the whole feature space, producing more confident classification results.
- We evaluate our framework on extensive benchmark datasets and the experimental results show that our method remarkably outperforms the existing state-of-the-art safe SSL methods.

2. Related Work

2.1. Semi-supervised Learning

Semi-supervised learning (SSL) has been a long-term research topic, and recently it has made a splash in computer vision (CV) attributed to the advancement of deep learning. The core idea of SSL is to make full use of extensive unlabeled data with as little labeled data as possible to achieve an approximate performance of fully supervised learning. There are a vast number of SSL methods proposed to resolve various CV tasks, which can be briefly categorized into pseudo-labeling, consistency regularization, and hybrid methods. The pseudo-labeling methods [3, 21, 15, 27] aim to produce pseudo labels for unlabeled data, and then supplement them into the labeled set to strengthen the model training. The consistency regularization methods [33, 29, 19, 24, 36, 35, 30, 13] assume that the decision boundary is supposed to cross the low-density region of samples, based on which they encourage the outputs of the unlabeled data and their perturbations to keep consistent in order to learn a discriminative decision boundary. Hybrid [1, 2, 32, 22] methods combine the two methods mentioned above and use some data augmentation approaches [5, 6, 39, 43] to enhance the SSL methods. However, these methods are based on the assumption that all labeled and unlabeled data share the same label space, which is often broken in real-world problems.

2.2. Safe Semi-Supervised Learning

In the open world, the unlabeled set is prone to contain some out-of-distribution (OOD) samples that do not belong to any seen class in the labeled dataset. It has been demonstrated that these OOD data will harm the in-distribution (ID) data classification on all seen classes [25]. Various solutions have been proposed to solve such a safety problem, called safe SSL methods [8, 4, 10, 11, 3, 41, 14]. A widely adopted way of safe SSL is to train a model on labeled data and use it to detect and remove the OOD data. Guo *et al*. [8] designed a weight function to subdue the weight of OOD data. Chen *et al*. [4] proposed a score function to filter out the OOD data based on their ensembled prediction results. However, by discarding the OOD data, these methods also abandon the valuable knowledge contained in them[10]. In contrast, Huang *et al*. [14] proposed a self-supervised warm-up training method to fully utilize the OOD data to further promote the OOD detection. Furthermore, He *et al*. [10] utilized the OOD data by calibrating their category distributions into uniform ones. Nevertheless, the application of OOD data is still dependent on the accuracy of the previous OOD data identification, which may be compromised by the limited labeled data. Different from the above methods, our work explicitly models all OOD data as a novel class peer to seen-classes, thus transferring the safe SSL from an open-set problem to close-set one. This brave attempt unifies the classification of OOD data and ID data into one feature space, enabling the full use of all OOD data and ID data and eliminating the possible negative impact of low performance when identifying OOD data separately.

2.3. Negative Learning

For SSL methods, incorrect supervisory information can cause catastrophic effects on network performance. The negative learning (NL) methods are therefore proposed to reduce the risk of incorrect information from the noisy labels in the labeled set [16], or the erroneously assigned pseudo labels in the unlabeled set [37, 38]. Generally, the routine NL is conducted by selecting a class randomly from all classes except for the given one as the complementary label per iteration, then optimizing the output probability corresponding to the complementary label to approach 0 [16]. Nevertheless, the existing NL methods only mine the useful

information on one complementary label, which has little impact on performance when the number of classes is large. Moreover, the complementary label could be variable as the training goes, bring much uncertainty to the model training. Our work addresses these problems by extending the vanilla NL to an iterative version, called INL. Our INL paradigm employs a memory bank to progressively update the complementary label until it covers most negative classes and remains one as the positive one. By employing the INL paradigm, our model can produce more confident classification results.

3. Methodology

3.1. Overview

Problem Formulation. Like the standard SSL problems, we assume that the training set of safe SSL contains two subsets: a labeled set $D^l = \{(x_i^l, y_i^l)\}_{i=1}^N$ with N samples and an unlabeled dataset $D^u = \{(x_i^u)\}_{i=1}^M$ with M samples, and $N \ll M$. Here, x_i^l or x_i^u represents the input image, $y_i^l \in \{1, 2, \cdots K\}$ denotes its corresponding label, and K is the number of seen classes. Notably, the safe SSL setting indicates that in the unlabeled set, besides x_i^u belonging to the K seen classes, or called ID data, there may be x_i^u that goes beyond the seen classes, called OOD data. The OOD data lead to a class distribution mismatch with ID data, which will further harm the performance of standard SSL methods. Our framework is proposed to ensure the safety and performance of the target classification task.

Framework Overview. As Figure 3 shows, our framework mainly contains two stages. The first stage aims to pre-train a prototype network on both labeled and unlabeled data, enabling the network to generate accurate prototypes of K seen classes (Section 3.2). Then, we employ a distance-based function to distinguish the OOD data from the ID data, and model the prototype of all OOD data, thus turning the K-way classification to the (K+1)-way classification (Section 3.3). By regarding the prototype of OOD data as new class peer to seen classes, we solve the safe SSL problem in a simpler close-set way in the second stage by utilizing the common SSL techniques (pseudo labeling and consistency regularization) (Section 3.4). To cope with the erroneously assigned pseudo labels, we propose an iterative negative learning (INL) paradigm to relieve the incorrect guidance and improve the classification performance (Section 3.5).

3.2. Prototype Network Pre-training

In our framework, we adopt a prototype network [31] for the classification task by generating a set of prototypes $C = c_1, \cdots, c_K$ as the anchors of seen classes. A good prototype network contributes to generating accurate seen class prototypes, or ID prototypes, which can lay solid foundations for the following OOD data identification. To achieve this, we pre-train the prototype network on both D^l and D^u to improve its ability of feature representation.

Learning from Labeled Data. In each training iteration, we can first acquire the ID prototype of class k ($k = 1, \ldots, K$) based on the labeled set D^l :

$$c_k = \frac{1}{|D_k|} \sum_{(x_i, y_i) \in D_k} f_\theta(x_i^l), \tag{1}$$

where D_k denotes the set of samples belonging to class k, f_θ is the prototype network parameterized by θ. The prototype c_k is actually the average of embedding features $f_\theta(x_i^l)$ belonging to class k. With these prototypes, we can obtain a K-dimensional vector that measures the similarity between $f_\theta(x_i^l)$ and the prototype set C:

$$d_i^l = \left(f_\theta(x_i^l) \cdot c_1, \cdots, f_\theta(x_i^l) \cdot c_K\right)^T, \tag{2}$$

where "$\cdot$" denotes the inner product. Finally, the probability distribution p_i^l of x_i^l regarding the K seen classes can be derived by applying a softmax function to the similarity vector d_i^l. To train the model, we use a standard cross-entropy loss on p_i^l and its corresponding label y_i^l as follows:

$$\mathcal{L}_{ce} = -\frac{1}{b^l} \sum_{i=1}^{b^l} \ln\left(p_i^l\left[y_i^l\right]\right), \tag{3}$$

where b^l denotes the number of labeled data in a mini-batch, and $p_i^l[k]$ represents the k-th element in p_i^l.

Learning from Unlabeled Data. To fully leverage the unlabeled data to enhance the representation capacity of the prototype network, we introduce an auxiliary self-supervised rotation prediction task with all unlabeled data. Specifically, given an unlabeled sample x_i^u, we rotate it by 0°, 90°, 180°, 270°, respectively, denoted as $x_{i,j}^u, j \in 1, 2, 3, 4$. Through the prototype network f_θ, we can obtain the embedding feature $f_\theta(x_{i,j}^u)$. Then the embedding feature is further sent to a 4-way rotation classification head h_φ parameterized by φ, which consists of a liner layer and a softmax layer, to produce the prediction $q_{i,j}^u = h_\varphi(f_\theta(x_{i,j}^u))$. Accordingly, we additionally add a rotation prediction loss in the pre-training stage:

$$\mathcal{L}_{rot} = -\frac{1}{4b^u} \sum_{i=1}^{b^u} \sum_{j=1}^{4} \ln\left(q_{i,j}^u\left[j\right]\right), \tag{4}$$

where b^u denotes the number of unlabeled data in a mini-batch. We use the $\mathcal{L}_{ce}$ and $\mathcal{L}_{rot}$ to optimize our model in stage 1.

Prototype Update. At the first training iteration, all the prototypes are initialized randomly. At the end of each iteration, we update all the ID prototypes based on the

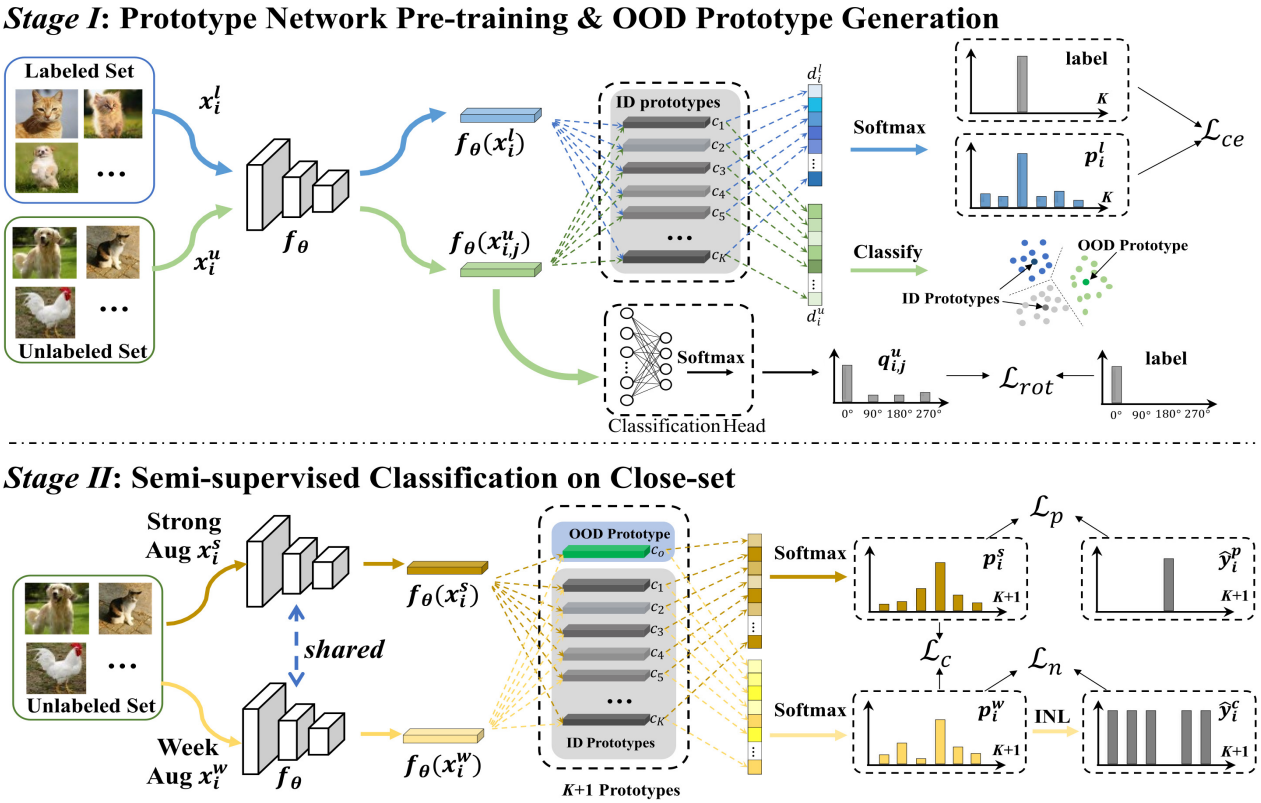

Figure 3. The overview of our framework, consisting of two stages. Stage I aims to pre-train the prototype network with labeled and unlabeled data by a cross-entropy loss $\mathcal{L}_{ce}$ and a rotation prediction loss $\mathcal{L}_{rot}$, respectively, then filter out the OOD data and model their prototype. Stage II shows the semi-supervised classification task on unlabeled data in a close-set way. The training losses include a consistency loss $\mathcal{L}_c$, a positive learning loss $\mathcal{L}_p$, and a negative learning loss $\mathcal{L}_n$. The loss on labeled data is omitted for simplicity.

trained prototype network by re-calculating them according to Equation 1. As the training goes, the accuracy of the prototypes will get improved, and reach a satisfying level when pre-training is over.

3.3. OOD Prototype Generation

After the pre-training, we have obtained a powerful prototype network and an ID prototype set C with regard to seen classes. Based on this, we can distinguish all the OOD data from the ID data in the unlabeled set. The core idea is to use a distance-based function to filter out those data far away from all the ID prototypes. Specifically, for each unlabeled sample x_i^u, we calculate its distance from all ID prototypes as follows:

$$d_i^u = \left(\|f_\theta\left(x_i^u\right) - c_1\|_2, \cdots, \|f_\theta\left(x_i^u\right) - c_K\|_2\right)^T. \quad (5)$$

Then we classify the x_i^u as ID or OOD data based on the minimal value in its distance vector d_i^u:

$$x_i^u \in \begin{cases} ID\ data, & \min(d_i^u) \leq \lambda \\ OOD\ data, & \min(d_i^u) > \lambda \end{cases}, \quad (6)$$

where λ is a threshold for OOD data filtering, which is determined by the OTSU algorithm [26]. For all the filtered OOD data, we can calculate their OOD prototype c_o as Equation 1 does. Thus, we now have K ID prototypes and a peer OOD prototype. The prototype set C can be updated by adding c_o, namely $C = C \cup \{c_o\}$, where $|C| = K + 1$. In this way, we successfully turn the K-way classification task on the open-set to the (K+1)-way classification task on the close-set. Correspondingly, for any input x_i, we can attain its probability distribution p_i over seen and unseen classes by computing the similarity between its embedding feature and the ID prototypes along with the OOD prototype, which just needs a small adjustment to Equation 2 by adding a term of $f_\theta(x_i) \cdot c_o$.

3.4. Close-set Semi-supervised Classification

Since we have K ID prototypes and an OOD prototype, we can perform (K+1)-way classification like solving a standard SSL problem. For the labeled data, we also employ the entropy loss of Equation 3 to optimize our framework. For the unlabeled data, inspired by FixMatch [32], we combine consistency regularization and pseudo-labeling to make full use of them. Particularly, for an input unlabeled sample x_i^u, we apply weak and strong data augmentations to get its two augmented versions x_i^w and x_i^s. Through the

prototype network and the softmax function, we can obtain two corresponding probability distributions p_i^w and p_i^s.

Consistency Regularization. The consistency regularization approach assumes that the predictions of a sample should remain consistent after different perturbations. Therefore, we impose a consistency loss on p_i^w and p_i^s to encourage them to be similar via KL divergence, which can be defined as follows:

$$\mathcal{L}_c = \frac{1}{b^u} \sum_{i=1}^{b^u} KL(p_i^w, p_i^s). \tag{7}$$

Pseudo-Labeling. Meanwhile, we regard the probability p_i^w from weak augmentation as pseudo label $\hat{y}_i^p$ for x_i^u. Considering the possible wrongly assigned pseudo labels, we propose to split the unlabeled data into high-confidence (HC) and low-confidence (LC) regions based on the maximum value in its pseudo label. For the HC data, we send them to positive learning, where pseudo labels are used to compute a positive loss with p_i^S:

$$\mathcal{L}_p = -\frac{1}{b^p} \sum_{i=1}^{b^p} \mathbb{1}\left(p_i^w[m] > \tau_1\right) ln\left(p_i^S\left[\hat{y}_i^p\right]\right), \tag{8}$$

where b denotes the number of HC data in a mini-batch, m is the index of the largest element in $\hat{y}_i^p$, τ_1 is a threshold to divide HF and LF data, and $\mathbb{1}(\cdot)$ is an indication function (equaling 1 if the condition in the bracket is satisfied, otherwise 0).

As for the LF data with $p_i^w[m] \leq \tau_1$, the network is prone to assign wrong pseudo labels. To mitigate the accumulation of incorrect information provided by the erroneous pseudo labels during training, we resort to negative learning (NL) [16] and propose an iterative negative learning (INL) paradigm, which will be introduced in the next subsection in detail.

3.5. Iterative Negative Learning

Conventional NL [16, 37] believes that although a network cannot confidently tell which class an LF data belongs to, it can learn knowledge from the class the data does not belong to, or called negative class. Nevertheless, conventional NL only selects one negative class as the complementary label, which only provides limited information for learning. In addition, conventional NL tends to produce wavering complementary labels, hindering the stable training of the network. To relieve these problems, we improve NL to an iterative version, called INL. Different from NL, INL maintains a memory bank $\mathcal{M}$ with the size of $M \times (K+1)$ to save the historical complementary labels $\{\hat{y}_i^c\}$ for all the LF data after each iteration. Based on the historical complementary label, we can iteratively update it to cover more negative classes in the next iteration. With all the more comprehensive complementary labels, our framework can be enhanced to classify more confidently.

Iterative Complementary Labels. The complementary label $\hat{y}_i^c$ in INL is updated in an iterative way. Specifically, in an iteration, given the probability distribution p_i^w from the sample x_i^w, we pick the lowest probability score and update the $\hat{y}_i^c$ in $\mathcal{M}_i$ $(i = 1, \ldots, M)$ as follows:

$$\hat{y}_i^c[l] = \mathcal{M}_i[l] = \begin{cases} 1, & p_i^w[l] \leq \tau_2 \\ 0, & p_i^w[l] > \tau_2 \end{cases}, \tag{9}$$

where $l = argmin\,(p_i^w)$ denotes the index of class with the lowest probability score, and τ_2 is the threshold to decide whether the class in $\hat{y}_i^c$ could be regarded as a negative class confidently. Once the complementary label $\hat{y}_i^c$ is updated, we save it in $\mathcal{M}_i$ and use it to optimize our framework via a negative learning loss. Then, the probability distribution p_i^w can be further calibrated in the next iteration. Accordingly, the complementary label $\hat{y}_i^c$ is iteratively updated by including a new negative class. Please note that, during the iterative process, if the highest probability score is over τ_1, the p_i^w will be utilized for positive learning as Equation 8 does. If the number of all negative classes reaches K, i.e., only one class remains uncertain, the iterative process will stop. We summarize this iterative process in Algorithm 1. Finally, the completely updated complementary label $\hat{y}_i^c$ serves for the negative learning loss in the following network training.

Negative Learning. Considering that the complementary label indicates the negative classes that a sample should not belong to, we encourage the prediction probability scores on these negative classes to approach 0. Consequently, we can perform the negative learning by the following negative learning loss:

$$\mathcal{L}_n = -\frac{1}{b^n} \sum_{i=1}^{b^n} \sum_{j=1}^{K+1} \hat{y}_i^c[j] \ln\left(1 - p_i^w[j]\right), \tag{10}$$

where b^n denotes the number of LF data in a mini-batch.

In summary, our framework is trained in stage two by a total loss as below:

$$\mathcal{L}_2 = \mathcal{L}_{ce} + \mathcal{L}_n + \alpha\left(\mathcal{L}_p + \mathcal{L}_c\right), \tag{11}$$

where α is a coefficient to balance these terms.

4. Experiments

4.1. Datasets

To validate the effectiveness of our proposed framework, we carry out experiments on the same public datasets as [10] for semi-supervised image classification, including MNIST [20], CIFAR-10 [17], CIFAR-100 [17] and Tiny-ImageNet [7]. To achieve the safe SSL setting, we select various ratios of unlabeled data from seen-classes, namely

Algorithm 1: Iterative Generation of Complementary Label

Input: The probability distribution $p_i^w \in R^{K+1}$ of sample x_i^u; the corresponding memory bank $\mathcal{M}_i \in R^{K+1}$.
Parameters: The number of negative classes n^-; the thresholds τ_1 and τ_2.
Initialize: n^-=0; τ_1=0.95; τ_2=0.05; a complementary label $\hat{y}_i^c \in R^{K+1}$ initialized by 0.
while $n^- < K$ **do**
  Obtain the index $l = \text{argmin}\,(p_i^w)$ and $\mathcal{M}_i[l] = 0$;
  if $\max\,(p_i^w) > \tau_1$ **then**
    compute the positive learning loss $\mathcal{L}_p$ based on Equation 8;
    break;
  end
  else
    if $p_i^w\,[l] \leq \tau_2$ **then**
      update $\hat{y}_i^c\,[l]$=1 and save it in $\mathcal{M}_i[l]$;
      n^-++;
    end
    compute the negative learning loss $\mathcal{L}_n$ based on Equation 10;
  end
end
Output: The updated complementary label $\hat{y}_i^c$.

mismatch ratio. For instance, when the mismatch ratio is 0.5, it means that half of the unlabeled data is from seen classes and the remaining is from unseen classes. *The detailed description of datasets can be found in the Supplementary Material.*

4.2. Implementation details

When training with MNIST, we employ a two-layer CNN as the backbone network using stochastic gradient descent (SGD) with a learning rate of $1e^{-3}$ [8]. The network is trained for 500 epochs with a batch size of 100. As for CIFAR-10, CIFAR-100, and TinyImageNet, the Wide-ResNet28-2 [42] is used as the backbone network. We use SGD to train the network with an initial learning rate of 0.03 which is adjusted via the cosine decay strategy, and a momentum of 0.9. The network in both stages is trained for 1024 epochs with a batch size of 64. The hyperparameter α in Equation 11 is dynamically updated as in [14]. *In addition, we also investigate the influence of two thresholds τ_1, τ_2 in the Supplementary Material.*

4.3. Comparison Methods

We compare our method with following state-of-the-art (SOTA) SSL methods: Pi-Model [29], Pseudo-Labeling (PL) [21], Temporal Ensembling [19], Mean Teacher [33], Virtual Adversarial Training (VAT) [24], FixMatch [32], Deep Safe Semi-Supervised Learning (DS3L) [8], Uncertainty Aware Self-Distillation (UASD) [4], Multi-Task Curriculum (MTC) [41], Curriculum Labeling (CL) [3], Safe Parameter Learning (SPL) [11], OpenMatch [28], Trash to Treasure (T2T) [14], and SAFE-STUDENT [10]. All the methods are compared on the testing set containing only ID samples. Note that, DS3L, UASD, MTC, CL, SPL, OpenMatch, T2T and SAFE-STUDENT are specially tailored for solving the safe SSL problem.

4.4. Experimental Results

4.4.1 Comparison with SOTA Methods

Quantitative Analysis. Table 1 reports the results more comprehensively on all datasets with 0.3 or 0.6 mismatch ratio. Besides the comparison on MNIST and CIFAR-10, particularly, we list the quantitative comparison results on two larger datasets CIFAR-100 and TinyImageNet in the third and fourth columns. As can be seen, when the mismatch ratio is 0.3, our method achieves 72.5% accuracy in average on CIFAR-100, about 4.1% higher than SAFE-STUDENT and 2.7% higher than T2T which performs best in all other methods. Similarly, for the harder TinyImageNet dataset, our method can still outperform other safe SSL methods by at least 2.0% average accuracy even with a high-level ratio of 0.6. All these comparison results demonstrate the effectiveness and robustness of our method in resolving the safe SSL problem under open-set assumption. *We also provide qualitative analysis in the Supplementary Material.*

4.4.2 Ablation Study

We validate the effectiveness of the proposed loss functions, i.e., $\mathcal{L}_c$, $\mathcal{L}_p$, $\mathcal{L}_n$, by ablating them and measuring the performance on CIFAR-100 with mismatch ratio of 0.3. The results are reported in Table 3. By comparing the first and second rows, we can know the $\mathcal{L}_c$ loss brings about 5.3% improvement. When integrating the $\mathcal{L}_p$ loss in the third row or the $\mathcal{L}_n$ loss in the fourth row, the performance gets further improved by 1.6% and 1.4%, respectively. With all the losses applied, our method can obtain the best accuracy of 72.5%. Through the ablation study, we can conclude that each proposed loss has a positive effect on our method.

4.4.3 Identification of Unseen-Class

Since the unseen-class of OOD samples plays an important role in our method, we further evaluate the unseen-class

Table 1. Seen-class classification accuracy (%) of different methods on the four datasets.

Method	MNIST		CIFAR-10		CIFAR-100		TinyImagenet	
	ratio=0.3	ratio=0.6	ratio=0.3	ratio=0.6	ratio=0.3	ratio=0.6	ratio=0.3	ratio=0.6
Pi-Model [29]	92.4±0.6	86.6±0.5	75.7±0.7	74.5±1.0	59.4±0.3	57.9±0.3	36.9±0.4	36.4±0.5
PL [21]	90.0±0.7	86.0±0.6	75.8±0.8	74.6±0.7	60.2±0.3	57.5±0.6	36.6±0.6	35.8±0.4
VAT [24]	94.5±0.3	90.4±0.3	76.9±0.6	75.0±0.5	61.8±0.4	59.6±0.6	36.7±0.5	36.3±0.6
FixMatch [32]	-	-	81.5±0.2	80.9±0.3	65.9±0.3	65.2±0.3	-	-
DS3L [8]	96.8±0.3	94.5±0.4	78.1±0.4	76.9±0.5	-	-	-	-
UASD [4]	96.2±0.6	94.3±0.8	77.6±0.4	76.0±0.4	61.8±0.4	58.4±0.5	37.1±0.7	36.9±0.6
MTC [41]	93.7±0.5	88.5±0.3	85.5±0.6	81.7±0.5	63.1±0.6	61.1±0.3	37.0±0.5	36.6±0.4
CL [3]	96.9±0.1	95.6±0.4	83.2±0.4	82.1±0.4	63.6±0.4	61.5±0.5	37.3±0.7	36.7±0.8
CL+SPL [10]	-	-	87.8±0.3	84.1±0.5	65.9±0.3	65.5±0.4	38.6±0.5	37.7±0.5
OpenMatch [28]	97.8±0.2	96.0±0.2	88.2±0.2	85.5±0.3	68.7±0.1	68.4±0.2	37.9±0.4	37.0±0.3
SAFE-STUDENT [10]	98.3±0.3	96.5±0.1	85.7±0.3	83.8±0.1	68.4±0.2	68.2±0.1	37.7±0.3	37.1±0.3
T2T [14]	98.4±0.1	96.2±0.2	89.0±0.4	86.9±0.2	69.8±0.2	68.0±0.2	39.1±0.3	37.3±0.3
Our-Method	**98.7±0.2**	**96.9±0.1**	**91.4±0.3**	**89.1±0.1**	**72.5±0.2**	**70.4±0.1**	**40.8±0.3**	**39.9±0.3**

Method	ratio=0	ratio=0.1	ratio=0.2	ratio=0.3	ratio=0.4	ratio=0.5	ratio=0.6	Avg
Probabilities	84.3 ± 0.9	84.3 ± 0.9	84.3 ± 0.9	84.3 ± 0.9	84.3 ± 0.9	84.3 ± 0.9	84.3 ± 0.9	84.3
DS3L	95.9 ± 0.8	93.1 ± 0.4	91.7 ± 0.2	90.6 ± 0.1	90.5 ± 0.5	89.1 ± 0.2	85.1 ± 0.8	90.9
SAFE-STUDENT	98.1 ± 0.1	97.3 ± 0.2	96.5 ± 0.1	96.0 ± 0.9	94.6 ± 0.9	93.5 ± 0.3	91.4 ± 0.2	95.3
Proposed	98.4 ± 0.1	97.8 ± 0.1	97.0 ± 0.2	96.5 ± 0.3	95.2 ± 0.5	94.0 ± 0.2	92.1 ± 0.3	95.9
Proposed(ID&OOD)	**98.7±0.3**	**98.0±0.2**	**97.5±0.1**	**97.0±0.2**	**96.1±0.4**	**94.8±0.2**	**93.0±0.2**	**96.4**

Table 2. AUC (%) for unseen-class identification on MNIST.

Method	CIFAR-100
our method w/o $\mathcal{L}c$, $\mathcal{L}p$ and $\mathcal{L}n$	64.7
our method w/o $\mathcal{L}p$,$\mathcal{L}n$	70.0
our method w/o $\mathcal{L}n$	71.6
our method w/o $\mathcal{L}p$	71.4
our method	**72.5**

Table 3. Seen-class classification accuracy (%) of ablation study on CIFAR-100 with mismatch ratio of 0.3.

identification ability of our method by comparing it with probability method [12], DS3L [8], SAFE-STUDENT [10] on MNIST in this experiment. Similar to [10], we also use AUC to evaluate the identification ability, in which we regard the unseen class data as a negative class and the others as a positive one. Table 2 lists the comparison results under different mismatch ratios. It can be found that the recent DS3L and SAFE-STUDENT lag behind our method by 5% and 0.6% accuracy, respectively. To improve the identification accuracy, we further make some attempts to modulate the way of splitting ID and OOD samples. We denote the improved version as proposed (ID&OOD). It clearly rises the average accuracy from 95.9% to 96.4%, proving the excellent ability of our method.

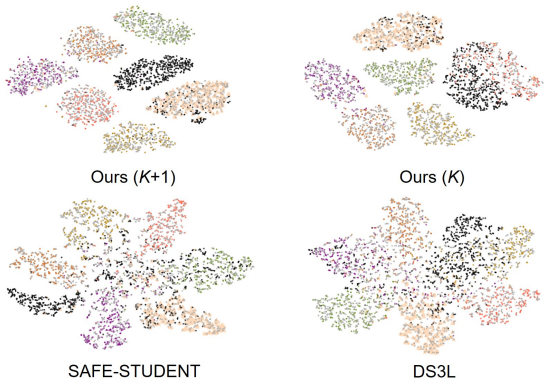

Figure 4. Visualization of feature distributions regarding different methods on CIFAR-10 via t-SNE. Black dots denote the OOD features, grey dots denote the unlabeled data features, and other colored dots denote the ID features.

4.4.4 Insight of Feature Distribution

To better support the effectiveness of our idea of turning N-way classification to (K+1)-way classification, we visualize the feature distributions of our method with (K+1)-way classification (denoted as Ours (K+1)), our method with K-way classification (denoted as Ours (K)), DS3L [8], and SAFE-STUDENT [10] by t-SNE [34] plotting. All the models are trained on CIFAR-10 (Fewer classes for bet-

ter presentation). As shown in Figure 4, although SAFE-STUDENT clusters the features of different classes in a more compact way than DS3L, our method can perform better by reducing the intra-class distance and increasing the inter-class distance. Without treating the OOD data as an individual class, the OOD features have already shown a tendency to gather. Once when we explicitly build the OOD prototype, all the classes, including the unseen-class marked in black, can be better distinguished, demonstrating the advancement of our powerful idea.

5. Conclusion

In this paper, we rethink the safe SSL problem from a novel aspect. Concretely, based on the experimental phenomenon that OOD data appear to cluster in the feature space, we propose a prototype-based framework to explicitly model the prototype of all OOD data. By treating the OOD prototype as equivalent to other ID prototypes, we transform the safe semi-supervised classification problem from open-set to close-set. Therefore, we can solve safe SSL using standard SSL techniques, including consistency regularization and pseudo-labeling. For the pseudo-labeling, we develop an INL paradigm to make full use of low-confidence pseudo labels by excavating more knowledge on wider classes. Extensive experiments on popular benchmarks demonstrate the effectiveness and superiority of our method.

6. Acknowledgement

This work is supported by National Natural Science Foundation of China (NSFC 62071314), Sichuan Science and Technology Program 2023YFG0263, 2023YFG0025, 2023NSFSC0497, and Opening Foundation of Agile and Intelligent Computing Key Laboratory of Sichuan Province.

References

[1] David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *Proceedings of the International Conference on Learning Representations*, 2020. 3

[2] David Berthelot, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5050–5060, 2019. 3

[3] Paola Cascante-Bonilla, Fuwen Tan, Yanjun Qi, and Vicente Ordonez. Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6912–6920, 2021. 3, 7, 8

[4] Yanbei Chen, Xiatian Zhu, Wei Li, and Shaogang Gong. Semi-supervised learning under class distribution mismatch. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3569–3576, 2020. 2, 3, 7, 8

[5] Ekin D. Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019. 3

[6] Ekin Dogus Cubuk, Barret Zoph, Jonathon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Advances in Neural Information Processing Systems*, 2020. 3

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 6

[8] Lan-Zhe Guo, Zhenyu Zhang, Yuan Jiang, Yu-Feng Li, and Zhi-Hua Zhou. Safe deep semi-supervised learning for unseen-class unlabeled data. In *International Conference on Machine Learning*, pages 3897–3906, 2020. 2, 3, 7, 8

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1

[10] Rundong He, Zhongyi Han, Xiankai Lu, and Yilong Yin. Safe-student for safe deep semi-supervised learning with unseen-class unlabeled data. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 14565–14574, 2022. 2, 3, 6, 7, 8

[11] Rundong He, Zhongyi Han, Yang Yang, and Yilong Yin. Not all parameters should be treated equally: Deep safe semi-supervised learning under class distribution mismatch. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6874–6883, 2022. 2, 3, 7

[12] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *Proceedings of the International Conference on Learning Representations*, 2017. 8

[13] Lin Hu, Jiaxin Li, Xingchen Peng, Jianghong Xiao, Bo Zhan, Chen Zu, Xi Wu, Jiliu Zhou, and Yan Wang. Semi-supervised npc segmentation with uncertainty and attention guided consistency. *Knowledge-Based Systems*, 239:108021, 2022. 3

[14] Junkai Huang, Chaowei Fang, Weikai Chen, Zhenhua Chai, Xiaolin Wei, Pengxu Wei, Liang Lin, and Guanbin Li. Trash to treasure: Harvesting OOD data with cross-modal matching for open-set semi-supervised learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8290–8299, 2021. 2, 3, 7, 8

[15] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5070–5079, 2019. 3

[16] Youngdong Kim, Junho Yim, Juseung Yun, and Junmo Kim. NLNL: negative learning for noisy labels. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 101–110, 2019. 2, 3, 6

[17] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1, 6

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1106–1114, 2012. 1

[19] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *Proceedings of the International Conference on Learning Representations*, 2017. 1, 3, 7

[20] Yann LeCun. The mnist database of handwritten digits. 1998. 6

[21] Dong-Hyun et al. Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, 2013. 3, 7, 8

[22] Junnan Li, Richard Socher, and Steven C. H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *Proceedings of the International Conference on Learning Representations*, 2020. 3

[23] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6738–6746, 2017. 1

[24] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993, 2019. 3, 7, 8

[25] Avital Oliver, Augustus Odena, Colin Raffel, Ekin Dogus Cubuk, and Ian J. Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pages 3239–3250, 2018. 2, 3

[26] Nobuyuki Otsu. A threshold selection method from gray level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9:62–66, 1979. 5

[27] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V. Le. Meta pseudo labels. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 11557–11568, 2021. 3

[28] Kuniaki Saito, Donghyun Kim, and Kate Saenko. Openmatch: Open-set semi-supervised learning with open-set consistency regularization. 34:25956–25967, 2021. 7, 8

[29] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 1163–1171, 2016. 3, 7, 8

[30] Yuang Shi, Chen Zu, Pinli Yang, Shuai Tan, Hongping Ren, Xi Wu, Jiliu Zhou, and Yan Wang. Uncertainty-weighted and relation-driven consistency training for semi-supervised head-and-neck tumor segmentation. *Knowledge-Based Systems*, 272:110598, 2023. 3

[31] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017. 2, 4

[32] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, 2020. 1, 3, 5, 7, 8

[33] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, pages 1195–1204, 2017. 1, 3, 7

[34] Laurens van der Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008. 1, 8

[35] Kaiping Wang, Yan Wang, Bo Zhan, Yujie Yang, Chen Zu, Xi Wu, Jiliu Zhou, Dong Nie, and Luping Zhou. An efficient semi-supervised framework with multi-task and curriculum learning for medical image segmentation. *International journal of neural systems*, 32(09):2250043, 2022. 3

[36] Kaiping Wang, Bo Zhan, Chen Zu, Xi Wu, Jiliu Zhou, Luping Zhou, and Yan Wang. Semi-supervised medical image segmentation via a tripled-uncertainty guided mean teacher model with contrastive learning. *Medical Image Analysis*, 79:102447, 2022. 3

[37] Yuchao Wang, Haochen Wang, Yujun Shen, Jingjing Fei, Wei Li, Guoqiang Jin, Liwei Wu, Rui Zhao, and Xinyi Le. Semi-supervised semantic segmentation using unreliable pseudo-labels. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4238–4247, 2022. 2, 3, 6

[38] Xiu-Shen Wei, He-Yang Xu, Faen Zhang, Yuxin Peng, and Wei Zhou. An embarrassingly simple approach to semi-supervised few-shot learning. *arXiv preprint arXiv:2209.13777*, 2022. 2, 3

[39] Qizhe Xie, Zihang Dai, Eduard H. Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*, 2020. 3

[40] Haiqin Yang, Shenghuo Zhu, Irwin King, and Michael R Lyu. Can irrelevant data help semi-supervised learning, why and how? In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 937–946, 2011. 1

[41] Qing Yu, Daiki Ikami, Go Irie, and Kiyoharu Aizawa. Multi-task curriculum framework for open-set semi-supervised learning. In *Proceedings of the European Conference on Computer Vision*, pages 438–454, 2020. 2, 3, 7, 8

[42] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *The British Machine Vision Conference*, 2016. 7

[43] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *Proceedings of the International Conference on Learning Representations*, 2018. 3

[44] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016. 2