On Pseudo-Labeling for Class-Mismatch Semi-Supervised Learning

Lu Han *hanlu@lamda.nju.edu.cn*
State Key Laboratory for Novel Software Technology, Nanjing University

Han-Jia Ye *yehj@lamda.nju.edu.cn*
State Key Laboratory for Novel Software Technology, Nanjing University

De-Chuan Zhan *zhandc@nju.edu.cn*
State Key Laboratory for Novel Software Technology, Nanjing University

Reviewed on OpenReview: *https://openreview.net/forum?id=tLG26QxoD8*

Abstract

When there are unlabeled Out-Of-Distribution (OOD) data from other classes, Semi-Supervised Learning (SSL) methods suffer from severe performance degradation and even get worse than merely training on labeled data. In this paper, we empirically analyze Pseudo-Labeling (PL) in class-mismatched SSL. PL is a simple and representative SSL method that transforms SSL problems into supervised learning by creating pseudo-labels for unlabeled data according to the model's prediction. We aim to answer two main questions: (1) How do OOD data influence PL? (2) What is the proper usage of OOD data with PL? First, we show that the major problem of PL is imbalanced pseudo-labels on OOD data. Second, we find that OOD data can help classify In-Distribution (ID) data given their OOD ground truth labels. Based on the findings, we propose to improve PL in class-mismatched SSL with two components – Re-balanced Pseudo-Labeling (RPL) and Semantic Exploration Clustering (SEC). RPL re-balances pseudo-labels of high-confidence data, which simultaneously filters out OOD data and addresses the imbalance problem. SEC uses balanced clustering on low-confidence data to create pseudo-labels on extra classes, simulating the process of training with ground truth. Experiments show that our method achieves steady improvement over supervised baseline and state-of-the-art performance under all class mismatch ratios on different benchmarks.

1 Introduction

Deep Semi-Supervised Learning (SSL) methods are proposed to reduce dependency on massive labeled data by utilizing a number of cheap, accessible unlabeled data. Pseudo-Labeling (PL) (Lee, 2013) is a widely used SSL method. PL is simple yet effective, which creates pseudo-labels according to predictions of the training model itself. Then SSL can be transformed into standard supervised learning. Other representative SSL methods include consistency regularization (Laine & Aila, 2017; Tarvainen & Valpola, 2017; Miyato et al., 2019), holistic methods (Berthelot et al., 2019; Sohn et al., 2020), and generative methods (Kingma et al., 2014). The recent development of SSL shows that these methods have achieved competitive performance to supervised learning methods.

However, these SSL methods achieve their good results based on an assumption that unlabeled data are drawn from the same distribution as the labeled data. This assumption can be easily violated in real-world applications. One of the common cases is that some unlabeled data come from *unseen classes*. As is illustrated in Figure 1, in image classification, we can collect a lot of unlabeled images from the internet but usually, they cover broader category concepts than labeled data. Oliver et al. (2018) have shown that under such class-mismatched conditions, the performance of traditional SSL methods is damaged. Several methods are proposed for class-mismatched SSL, including filtering out OOD data (Yu et al., 2020; Chen et al., 2020), down weighting OOD data (Chen et al., 2020), and re-using OOD data by neural style transfer (Luo et al., 2021)/self-supervised learning (Huang et al., 2021). Although these methods achieve good results, why OOD data damage performance and how will OOD data help remain unclear.

In this paper, we focus on empirically analyzing one representative family of the SSL method — PL in class-mismatched SSL and give some answers to these two questions. (1) How do OOD data influence PL? (2) What are suitable pseudo-labels for OOD data? For question (1), we investigate pseudo-labels created by PL. The main finding is that pseudo-labels on OOD data tend to be imbalanced while on ID data, they remain balanced. We further show that PL's performance is damaged due to such an imbalance in OOD data. For question (2), several strategies for labeling OOD data are investigated. We conclude that it is beneficial when labeling OOD data as a class different from ID data, and the performance can be further improved when pseudo-labels partition unlabeled OOD data into their semantic clusters.

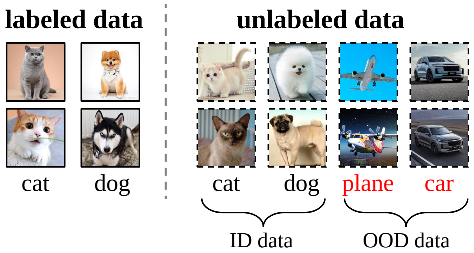


Figure 1: Realistic Semi-Supervised Learning may simultaneously contain unlabeled ID and OOD data. ID data come from the same classes as labeled data while OOD data come from classes that are not seen in labeled data.

Based on the experimental analyses, we propose a two-branched model called Υ-Model, which processes unlabeled data according to their confidence score on ID classes. The first branch performs Re-balanced Pseudo-Labeling (RPL) on high-confidence data. It utilizes the property of imbalanced pseudo-labels on OOD data, truncating the number of pseudo-labeled data for each class to their minimum. This procedure filters out many OOD data and also prevents the negative effect of imbalanced pseudo-labels. For the other branch, Semantic Exploration Clustering (SEC) is performed on low-confidence data. They are considered OOD data and their semantics will be mined by clustering into different partitions on extra classes. The clustering result provides better pseudo-labels for these OOD data than vanilla PL. Experiments on different SSL benchmarks show that our model can achieve steady improvement in comparison to the supervised baseline. Our contributions are:

- We analyze PL for ID and OOD data. The findings lead to two primary conclusions: (1) Imbalance of pseudo-labels on OOD data damages PL's performance. (2) Best pseudo-labels for unlabeled OOD data are those different from ID classes and partitioning them into their semantic clusters.
- We propose our two-branched Υ-Model. One branch re-balances pseudo-labels on ID classes and filters out OOD data. The other branch explores the semantics of OOD data by clustering on extra classes.
- Experiments on different SSL benchmarks empirically validate the effectiveness of our model.

2 Preliminary

2.1 Class-Mismatched SSL

Similar to the SSL problem, the training dataset of the class-mismatched SSL problem contains n ID labeled samples $\mathcal{D}_l = \{(\boldsymbol{x}_{li}, y_{li})\}_{i=1}^{n}$ and m unlabeled samples $\mathcal{D}_u = \{\boldsymbol{x}_{ui}\}_{i=1}^{m}$, (usually, $m \gg n$,) $y_{li} \in \mathcal{Y}_{ID} = \{1, \ldots, K_{ID}\}$, while different from SSL, the underlying ground truth $\mathbf{y}_u$ of unlabeled data may be different from labeled data. *i.e*, $y_{uj} \in \mathcal{Y}_{ID} \cup \mathcal{Y}_{OOD}, \mathcal{Y}_{OOD} = \{K_{ID}+1, \ldots, K_{ID}+K_{OOD}\}$. The goal of class-mismatched SSL is to **correctly classify ID samples into** $\mathcal{Y}_{ID}$ using labeled set with ID samples and unlabeled set possibly with OOD samples.

2.2 Pseudo-Labeling

Pseudo-Labeling (PL) leverages the idea that we can use the model itself to obtain artificial labels for unlabeled data (Lee, 2013). PL first performs supervised learning on labeled data to get a pre-trained model f, which outputs the probability of belonging to each ID class. Given $c(\boldsymbol{x})$ is the confidence score for $\boldsymbol{x}$

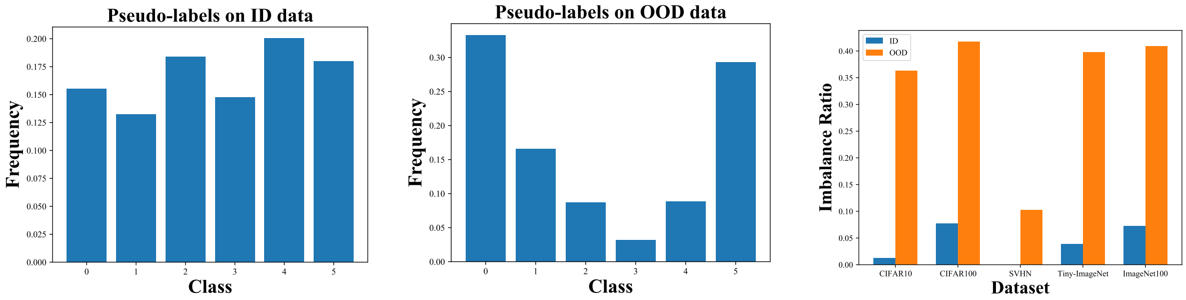
$$c(\boldsymbol{x}) = \max_{y \in \mathcal{Y}_{ID}} f(y|\boldsymbol{x}), \tag{1}$$

PL creates the pseudo-labels for each unlabeled sample:

$$y' = \begin{cases} \arg\max_{y \in \mathcal{Y}_{ID}} f(y|\boldsymbol{x}) & , \quad c(\boldsymbol{x}) > \tau \\ \text{reject} & , \quad \text{otherwise} \end{cases}, \tag{2}$$

All pseudo-labeled unlabeled data will be treated as labeled data for the next supervised learning generation. PL iteratively performs supervised learning and pseudo-label creation until stops.

3 Analysis of Pseudo-Labeling in Class-Mismatched SSL



(a) Pseudo-labels on ID (b) Pseudo-labels on OOD (c) Imbalance ratio

Figure 2: Analysis of the pseudo-label distributions created by the pre-trained model. (a) On ID data, pseudo-label distribution is balanced since they share the same distribution with the labeled data. (b) On OOD data, the pseudo-label distribution is *imbalanced.* (c) The imbalance ratio of pseudo-labels on ID and OOD data on other datasets. The imbalance ratio is computed by KL-divergence with uniform distribution.

In class-mismatched SSL, vanilla PL can only create pseudo-labels on ID classes even for OOD data. We will analyze how these OOD data influence vanilla PL and what are the better pseudo-labels for them in this section. Experiments are carried out on totally five kinds of datasets. (We use (n/m) to represent n ID classes and m OOD classes.)

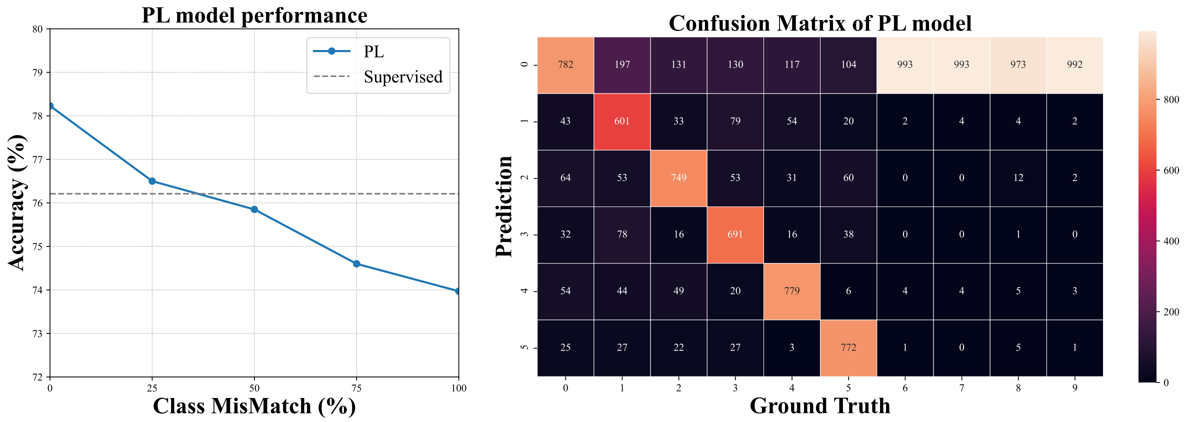
- **CIFAR10 (6/4)** : created from **CIFAR10** (Krizhevsky & Hinton, 2009).It takes the 6 animal classes as ID classes and 4 vehicle classes as OOD classes. We select 400 labeled samples for each ID class and totally 20,000 unlabeled samples from ID and OOD classes.
- **SVHN (6/4)**: We select the first "0"-"5" as ID classes and the rest as OOD. We select 100 labeled samples for each ID class and totally 20,000 unlabeled samples.
- **CIFAR100 (50/50)**: created from **CIFAR100** (Krizhevsky & Hinton, 2009). The first 50 classes are taken as ID classes and the rest as OOD classes. We select 100 labeled samples for each ID class and a total of 20,000 unlabeled samples.
- **Tiny ImageNet (100/100)**: created from **Tiny ImageNet**, which is a subset of **ImageNet** (Deng et al., 2009) with images downscaled to 64 × 64 from 200 classes. The first 100 classes are taken as ID classes and the rest as OOD classes. We select 100 labeled samples for each ID class and 40,000 unlabeled samples.
- **ImageNet100 (50/50)**: created from the 100 class subset of ImageNet (Deng et al., 2009). The first 50 classes are taken as ID classes and the rest as OOD classes. We select 100 labeled samples for each ID class and a total of 20,000 unlabeled samples.

Here we use C to represent CIFAR, TIN to represent Tiny ImageNet, IN to represent ImageNet for short. We vary the ratio of unlabeled images to modulate class distribution mismatch. For example, the extent is 50% means half of the unlabeled data comes from ID classes and the others come from OOD classes. We use Wide-ResNet-28-2 (Zagoruyko & Komodakis, 2016) as our backbone. We also adopt data augmentation techniques including random resized crop, random color distortion and random horizontal flip. For each epoch, we iterate over the unlabeled set and random sample labeled data, each unlabeled and labeled mini-batch contains 128 samples. We adopt Adam as the optimization algorithm with the initial learning rate 3×10^{-3} and train for 400 epochs. Averaged accuracies of the last 20 epochs are reported, pretending there is no reliable (too small) validation set to perform early stop (Oliver et al., 2018).

3.1 Imbalance of Pseudo-labels on OOD Data

In this section, we analyze the **pre-trained model** that creates the first set of pseudo-labels, and the **final model** trained by Pseudo-Labeling.

Pre-trained model. Like what is concluded in OOD detection (Hendrycks & Gimpel, 2017), ID data tend to have higher confidence than OOD data, but there are still considerable OOD data with high confidence. In class-mismatched SSL, the unlabeled data are in much larger quantities. When the class mismatch ratio is large, there are quite a few OOD data with high confidence scores. We will show in the final model experiments that these high-confidence OOD data damage performance. Secondly, we study pseudo-labels on both ID data and OOD data. Figure 2(a) shows that pseudo-labels ID data is balanced. However, they are rather imbalanced on OOD data (Figure 2(b)). Such difference in created pseudo-labels is attributed to the different distribution they are drawn from. Samples with certain pattern bias to certain classes. ID data bias to ID classes uniformly because they are sampled by the same distribution. However, with little probability, OOD data will also bias to ID classes uniformly since they have little relevance to ID data.[1]



(a) Performance of PL

(b) Confusion Matrix of PL model

Figure 3: (a) PL model degrades as the mismatch ratio increases. (b) Confusion matrix of PL model when the mismatch ratio = 100%. It demonstrates that the imbalance of pseudo labels on OOD data affects PL's performance. A lot of ID samples with class 1-5 are misclassified into class 0. Also, as the PL process continues, the imbalance of pseudo-labels on OOD data get even worse.

Final Pseudo-Labeling model. As an old saying goes, a good beginning is half done. However, such imbalance of the first set of pseudo-labels starts PL model badly when there is a large portion of OOD data, putting the model in danger of imbalanced learning. We run vanilla PL and show that the *imbalance of pseudo-labels harms the performance.* Figure 3(a) shows the performance of PL model with different OOD ratios. In accord with (Oliver et al., 2018), PL model degrades as the portion of OOD data gets larger. Figure 3(b) displays the confusion matrix of the PL model on the whole test set containing both ID and OOD data. Since only 6 classes are known to us, the confusion matrix is a rectangle. We can see almost

[1] How OOD data are generated affects the imbalance ratio. We experiment on different OOD class settings in Appendix C.2 to show that imbalanced pseudo-labels on OOD data is a general phenomenon for non-curated natural datasets.

all the OOD samples (class 6-9) are classified as class 0, which means the imbalance effect on OOD data gets even worse as the PL training goes on. The possible reason is that, unlike Pseudo-Labeling on ID data, supervision of labeled data can not help correct pseudo-labels on OOD data. Thus the imbalance continuously deteriorates. The imbalance on OOD data also influences classification performance on ID data. Samples of major classes (class 0) overwhelm the loss and gradient, leading to a degenerate model (Lin et al., 2017). We can see the PL model mistakenly classifies many of data with class 1-5 into class 0.

3.2 Pseudo-Labeling Strategy for OOD data

The previous section shows OOD data hurt the performance of vanilla PL. Then here comes the question: Assuming that we already know which data are OOD, **how do we use these OOD data? Is omitting them the best way? If not, what are the better pseudo-labels for them?** To answer these questions, we investigate four strategies to create pseudo-labels for OOD data:

- **Baseline.** This baseline omits all the OOD data and only trains on the labeled ID data.
- **Re-Assigned Labeling[2].** This strategy assigns data of each OOD class to an ID class. It ensures that different OOD class is assigned to different ID class, keeping the semantics unchanged between OOD classes. For example, (ship, trunk, airline, automobile) can be assigned to (bird, cat, deer, dog). This strategy can be seen as training a classifier of "super-classes".
- **Open-Set Labeling.** This strategy is named after the related setting – Open-Set Recognition (Scheirer et al., 2013; Bendale & Boult, 2016). This strategy treats all OOD data as one unified class $K_{ID}+1$. Thus this model outputs probability over $K_{ID}+1$ classes.
- **Oracle Labeling.** This strategy uses the ground truth of OOD data. Thus this model outputs probability over $K_{ID}+K_{OOD}$ classes.

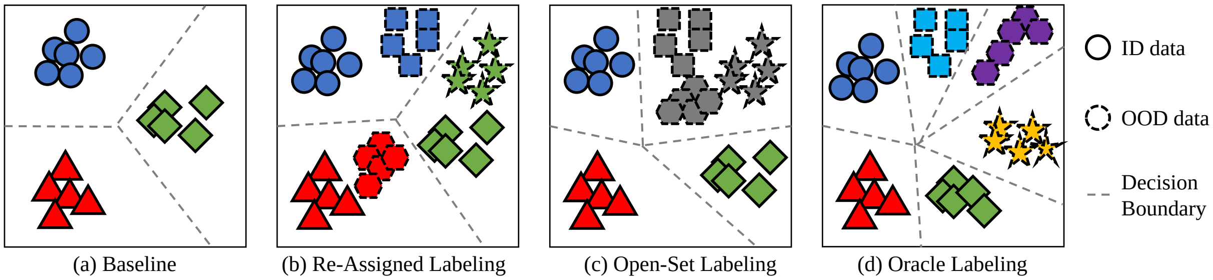
Note that Open-Set Labeling and Oracle Labeling can classify samples into more than K_{ID} classes. However, during evaluation, we only classify samples into K_{ID} ID classes. For these models, the predicted label $\hat{y}$ of a test sample $\boldsymbol{x}$ is calculated as:

$$\hat{y}(x) = \underset{y \in \mathcal{Y}_{ID}}{\arg\max}\, f(y|\boldsymbol{x}) \tag{3}$$

the overall comparison of the four strategies is illustrated in Figure 4. We also report test accuracy on the five datasets when the class-mismatched ratio is 100% in Table 1. From the results, we can get several important conclusions. (1) Re-Assigned Labeling underperforms baseline a little. This indicates that assigning samples with OOD classes to ID classes does not help the model distinguish between ID classes even if we somehow know which OOD data are semantically different. It also reveals that performing vanilla PL on OOD data may never help even if we do it perfectly. (2) Open-Set Labeling outperforms baseline, which indicates it improves the performance if we label the OOD data as a class other than ID classes. (3) We can see Oracle Labeling improves performance and achieves the best results among the four strategies. It means that in addition to labeling OOD data as extra classes, if we can further assign OOD data with different semantics to different classes, the model will achieve better results.

Discussion. Why does Oracle Labeling consistently outperform Open-Set Labeling by a large margin? We think the most important reason is that Oracle Labeling utilizes information among OOD classes. For example, in the experiment of CIFAR10(6/4), open-set labeling takes all the vehicle samples as one class, ignoring the fact that they come from four categories – airplane, automobile, ship and truck. The difference among data of these classes may provide useful information not covered by labeled data of ID classes, especially when the labeled data is not plenty. For example, learning to distinguish between airplane and truck helps distinguish between bird and dog by judging whether having wings or not. But the model trained by open-set labeling loses such benefit. In contrast, Oracle labeling can help the model capture this information by utilizing the ground truth labels. Consequently, oracle labeling performs better than open-set labeling.

[2] Note that Re-Assigned Labeling has many possibilities. If there are n ID classes and m OOD classes, A_n^m possible assignments exist. It is impossible to experiment on all of them. To deal with it, we randomly choose 10 possible assignment and pick the maximum performance among them.



(a) Baseline (b) Re-Assigned Labeling (c) Open-Set Labeling (d) Oracle Labeling

Figure 4: Four strategies of how to label OOD data. Different shapes represent different ground truths. Data with the same color are labeled as the same classes. A shape with a solid outline means it is an ID sample while OOD data are represented with a dashed line. (a) No Labeling acts as a baseline where OOD data are omitted. (b) Re-Assigned Labeling re-labels OOD data to certain ID classes. (c) Open-Set Labeling labels all the OOD data as a unified class. (d) Oracle Labeling uses the ground truths of OOD data.

Table 1: Performance of five different pseudo-labeling strategies on different datasets. It can be concluded: (1) Re-Assigned Labeling underperforms baseline. (2) Open-Set Labeling outperforms baseline a little. (3) Oracle Labeling improves performance and achieves the best results.

	C10 (6/4)	**SVHN (6/4)**	**C100 (50/50)**	**TIN (100/100)**	**IN (50/50)**
Baseline	76.21 (-)	88.33 (-)	58.68 (-)	39.08 (-)	48.12 (-)
Re-Assigned	75.99(-0.22)	84.40(-3.93)	50.52(-8.16)	34.90(-4.76)	45.60 (-2.52)
Open-Set	77.95(+1.74)	88.31(-0.02)	58.76(+0.08)	40.06(+0.86)	49.56(+1.44)
Oracle	79.25(+3.04)	92.07(+3.74)	63.90(+3.73)	45.28(+6.78)	55.96(+7.84)

3.3 Summary of Section

In this section, we study the behavior of the Pseudo-Labeling model in class-mismatched SSL. We summarize several important conclusions here:

Conclusion 1: Classification model trained with labeled ID data creates imbalanced pseudo-labels on OOD data while balanced pseudo-labels on ID data.

Conclusion 2: The vanilla PL makes the imbalance of pseudo-labels deteriorate, damaging the classification performance on ID data.

Conclusion 3: Labeling OOD data as ID classes does not help and may even perform worse.

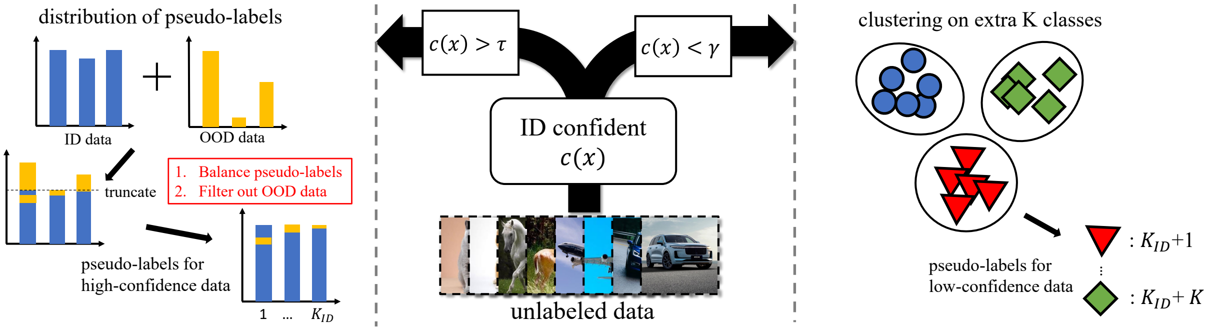
Conclusion 4: It is beneficial to label OOD data as extra classes different from ID classes. If we can further label semantically different OOD data as different classes, the performance can be further improved.

4 Method

Based on the findings in Section 3, we proposed Υ-Model (named after its shape) for class-mismatched SSL. Υ-Model trains a classifier f that will output the posterior distribution over $K_{ID} + K$ classes, *i.e*, $f(\mathbf{y}|\boldsymbol{x}) \in \mathbb{R}^{K_{ID}+K}, 1^\top f(\mathbf{y}|\boldsymbol{x}) = 1$. K is the number of extra classes, which can be known in advance (*i.e*, $K = K_{OOD}$) or be set as a *hyper-parameter*. Similar to vanilla PL, we define confidence with the same form as Equation 1. However, this confidence is a little different from its original definition in Hendrycks & Gimpel (2017), because we only calculate the maximum probability of the K_{ID} classes instead of all. Therefore, we rename it to **In-Distribution confidence (ID confidence)**. For evaluation, we predict labels using Equation 3. Υ-Model aims to solve the following questions:

Problem 1: how to avoid imbalanced pseudo-labels in PL model? (Conclusion 1, 2)

Problem 2: how to avoid labeling OOD data as ID? (Conclusion 3)



(a) Re-balanced Pseudo-Labeling (b) Υ-model (c) Semantic Exploration Clustering

Figure 5: Illustration of Υ-Model and its two main branches. (b) is the main structure of Υ-Model where we judge by the ID confidence if certain unlabeled data belongs to ID classes or not. The data with high confidence will perform Re-balanced Pseudo-Labeling, while those with low confidence will get their pseudo-labels by Semantic Exploration Clustering. (a) Re-balanced Pseudo-Labeling truncates the number of pseudo-labeled data to the minimum, making the pseudo-labels balanced and filtering out OOD data. (c) Semantic Exploration Clustering simulates the process of learning from ground truth labels of OOD data, creating pseudo-labels on extra K classes by clustering.

Problem 3: how to create proper pseudo-labels for unlabeled OOD data? (Conclusion 4)

Υ-Model consists of two main branches – Re-balanced Pseudo-Labeling (RPL) and Semantic Exploration Clustering (SEC). RPL acts on high-confidence data to solve Problem 1, 2. SEC acts on low-confidence data to solve Problem 3. We describe the two branches in the following sections. The overview of Υ-Model is illustrated in Figure 5.

4.1 Re-balanced Pseudo-Labeling

As illustrated in Section 3.2, the main problem of vanilla PL is that a large number of OOD data with high confidence scores have imbalanced pseudo-labels. One possible solution is re-weighting the unlabeled sample (Guo et al., 2020) or using other methods in the imbalance learning field. However, even if we solve the problem of imbalance learning, labeling OOD data as ID classes also may damage the performance (Conclusion 3). In this paper, we use a simple method – Re-balanced Pseudo Labeling – to simultaneously solve imbalance (Problem 1) and incorrect recognition (Problem 2). It produces a set $\mathcal{P}$ of pseudo-labeled samples in three steps:

$$N = \min_{y \in \mathcal{Y}_{ID}} |\{\boldsymbol{x} \in \mathcal{D}_u \mid f(y \mid \boldsymbol{x}) > \tau\}|, \tag{4}$$

$$\tau_y = \text{select\_N-th}(\{f(y \mid \boldsymbol{x}) \mid \boldsymbol{x} \in \mathcal{D}_u\}), \tag{5}$$

$$\mathcal{P} = \bigcup_{y \in \mathcal{Y}_{ID}} \{(\boldsymbol{x}, y) \mid f(y \mid \boldsymbol{x}) \geq \tau_y, \boldsymbol{x} \in \mathcal{D}_u\}, \tag{6}$$

where select_N-th returns the N-th biggest value of the given set. RPL first calculates the minimum number of pseudo-labeled samples for each ID class by Equation 4. Then it truncates the number of pseudo-labels of each ID class to that number by Equation 5, 6. The process of RPL is illustrated in Figure 5(a). First, it enforces the pseudo labels on ID classes to be balanced, solving Problem 1. Second, as is shown in Section 3, the set of high-confidence data is a mixture of ID and ODD data. Due to Conclusion 1, the pseudo-label distribution of such a set is a sum of imbalanced and balanced ones, thus still imbalanced. However, by selecting only top-N confident samples for each ID class, we will keep ID data and omit many OOD data since confidence on ID data tends to be higher than OOD data (Hendrycks & Gimpel, 2017). This process solves Problem 2.

4.2 Semantic Exploration Clustering

As is demonstrated in Section 3.2, if we know a set of samples is OOD, it will improve the performance if we label them as a unified class $K_{ID}+1$. But the best way is to use their ground truths (Conclusion 4). However, it is impossible to access their ground truths since they are unlabeled. We resort to using Deep Clustering methods (Caron et al., 2018; Asano et al., 2020) to mine their semantics and approximate the process of learning with the ground truths. Modern Deep Clustering can learn semantically meaningful clusters and achieves competitive results against supervised learning (Gansbeke et al., 2020). Here, we use the balanced clustering method in Asano et al. (2020); Caron et al. (2020) to create pseudo-labels for these OOD data. Assuming there are M samples recognized as OOD, we first compute their soft targets:

$$\begin{aligned} &\min_{Q \in U(K,M)} \langle Q, -\log P \rangle, \\ &U(K,M) := \left\{ Q \in \mathbb{R}_+^{K \times M} \mid Q1 = \frac{1}{K}1, Q^\top 1 = \frac{1}{M}1 \right\}, \end{aligned} \tag{7}$$

where $P \in \mathbb{R}_+^{K \times M}, P_{ij} = \hat{f}(K_{ID} + i|\boldsymbol{x}_j)$. $\hat{f}$ is the normalized posterior distribution on extra classes, *i.e*, $\hat{f}(K_{ID} + i|\boldsymbol{x}_j) = f(K_{ID} + i|\boldsymbol{x}_j)/\sum_{k=1}^{K} f(K_{ID} + k|\boldsymbol{x}_j)$. We use *the Sinkhorn-Knopp algorithm* (Cuturi, 2013) to optimize Q. Once we get Q, we harden the label by picking the class with the maximum predicted probability and mapping it to the extra K classes:

$$\hat{y}_j = K_{ID} + \arg\max_i Q_{ij}. \tag{8}$$

$\hat{y}_j$ is used as the pseudo-label for $\boldsymbol{x}_j$. We perform SEC on the set of data with ID confidence lower than a threshold γ, *i.e*, $\{\boldsymbol{x}|c(\boldsymbol{x}) < \gamma\}$. It may be the concern that introducing a clustering component makes the Υ-Model too computationally expensive to be practical. We give analyses of time complexity in Appendix F.

5 Related Work

Class-Mismatched Semi-Supervised Learning. Deep Semi-Supervised Learning suffers from performance degradation when there are unseen classes in unlabeled data (Oliver et al., 2018). As the proportion of such out-of-distribution (OOD) data get larger, the performance drop more. To cope with such a class-mismatched problem, several methods are proposed. Chen et al. (2020) formulate a sequence of ensemble models aggregated accumulatively on-the-fly for joint self-distillation and OOD filtering. Guo et al. (2020) re-weight the unlabeled data by meta-learning to decrease the negative effect of OOD data. Huang et al. (2020) recycle transferable OOD data employing adversarial learning. Recently,Saito et al. (2021) proposed open-set consistency regularization to improve outlier detection. Cao et al. (2021) proposed open-world semi-supervised learning, where the classes of unlabeled data need to be discovered. From a methodological perspective, both Cao et al. (2021) and our method use cluster to mining semantics. However, their method is originated from the demand of discovering novel classes, while ours is based on analyses of best labeling strategies. ORCA does not give such important analyses. Additionally, we solve the imbalanced pseudo-labels by RPL while they do not reveal this problem. Different from all these methods, we conduct a comprehensive study on Pseudo-Labeling (PL) and give useful guidance on how to do better in class-mismatched SSL. In addition to that, we reveal the imbalance phenomenon and propose RPL.

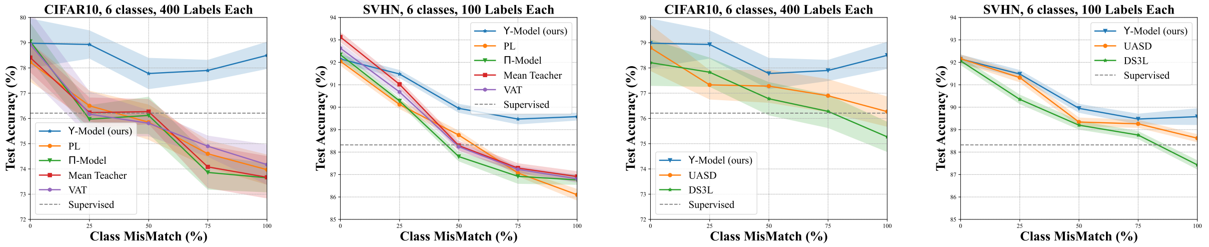
Pseudo-Labeling. The method of Pseudo-Labeling, also known as self-training, is a simple and effective way for Deep SSL (Lee, 2013; Shi et al., 2018; Arazo et al., 2020; Iscen et al., 2019). Despite its simplicity, it has been widely applied to diverse fields such as image classification (Xie et al., 2020), natural language processing (He et al., 2020) and object detection (Rosenberg et al., 2005). The use of a hard label makes Pseudo-Labeling closely related to entropy minimization (Grandvalet & Bengio, 2004).

Deep Clustering and Novel Class Discovery. Deep clustering methods improve the ability of traditional cluster methods by leveraging the representation power of DNNs. A common means is to transform data into low-dimensional feature vectors and apply traditional clustering methods (Yang et al., 2017; Caron et al., 2018). In Self-Supervised Learning, clustering methods are used to learn meaningful representation for downstream tasks (Caron et al., 2018; Asano et al., 2020; Caron et al., 2020). Modern Deep Clustering can learn semantically meaningful clusters and achieves competitive results against supervised learning (Gansbeke et al., 2020). With the help of Deep Clustering, the concept of Novel Class Discovery (NCD) was first formally introduced by Han et al. (2019). There are two main differences between NCD and our setting. First, the goal of NCD is to correctly cluster OOD data while semi-supervised learning aims to correctly classify ID data. Second, NCD knows which data are OOD in advance while class-mismatched SSL does not. Therefore, *NCD methods can not directly apply to this setting.* A recent method, UNO [1], uses a similar cluster method to discover semantics among OOD data like our SEC component. However, since they mainly focus on cluster OOD data, they do not reveal the benefit for classifying ID data when mining the semantics of OOD data, which is one of the contributions of this paper.

6 Experiments

Dataset. We test our methods on the five datasets as in Section 3, *i.e*, **CIFAR10 (6/4)**, **SVHN (6/4)**, **CIFAR100 (50/50)**, **Tiny ImageNet (100/100)** and **ImageNet100 (50/50)**. The class-mismatched ratio is set as $\{0\%, 25\%, 50\%, 75\%, 100\%\}$.

Implementation Details. We use the same network and training protocol as Section 3. We first train a classification model only on labeled data for 100 epochs without RPL and SEC. We update pseudo-labels every 2 epochs. For both datasets, we set $\tau = 0.95$, $\gamma = 0.3$,$K = 4$. We use an exponential moving average model for final evaluation as in Athiwaratkun et al. (2019).



(a) (b) (c) (d)

Figure 6: Comparison with existing methods on CIFAR10 and SVHN dataset with Wide-ResNet-28-2 network. Class mismatch ratios are varied. The shaded regions with the curves indicate the standard deviations of the accuracies over five runs. (a) (b) Comparison with traditional SSL methods. These methods suffer from performance degradation as the mismatch ratio increases. (c) (d) Comparison to two existing class-mismatched SSL methods – UASD and DS^{3}L. Our methods perform better in almost all the experimental setups.

6.1 Compare with Traditional SSL methods

In this subsection, we compare our methods with four traditional SSL methods – Pseudo-Labeling (Lee, 2013), Π-Model (Laine & Aila, 2017), Mean Teacher (Tarvainen & Valpola, 2017) and VAT (Miyato et al., 2019). [3] Figure 6(a), 6(b) show the results. Traditional methods suffer from performance degradation as the mismatch ratio increases. They usually get worse than the supervised baseline when the mismatch ratio is larger than 50% on CIFAR10 and SVHN. In contrast, our methods get steady improvement under all class mismatch ratios. The reasons can be attributed as follows. First, our method is aware of the existence of OOD data. We do not treat OOD data like ID data, which can hurt performance. Second, we reuse OOD data by exploring their semantics which proves to be useful in Section 3.2. Therefore, even when the class-mismatched ratio gets 100%, the performance of Υ-Model is still better than the supervised baseline.

6.2 Compare with Class-Mismatched SSL methods

In this subsection, we compare our method with two existing class-mismatched SSL methods – UASD (Chen et al., 2020) and DS^{3}L (Guo et al., 2020). For a fair comparison, we use Pseudo-Labeling as the base method of DS^{3}L. From Figure 6(c), Figure 6(d), we can see our methods are superior to these two methods in all settings. It is noticeable that DS^{3}L underperforms supervised baseline when all the unlabeled data are drawn

[3] We note that there are also other methods like FixMatch (Sohn et al., 2020) and MixMatch (Berthelot et al., 2019) focus on augmentation or other tricks. Their contribution is orthogonal to ours. Also, FixMatch is unstable in this setting. We provide comprehensive comparisons with these methods in Appendix D.

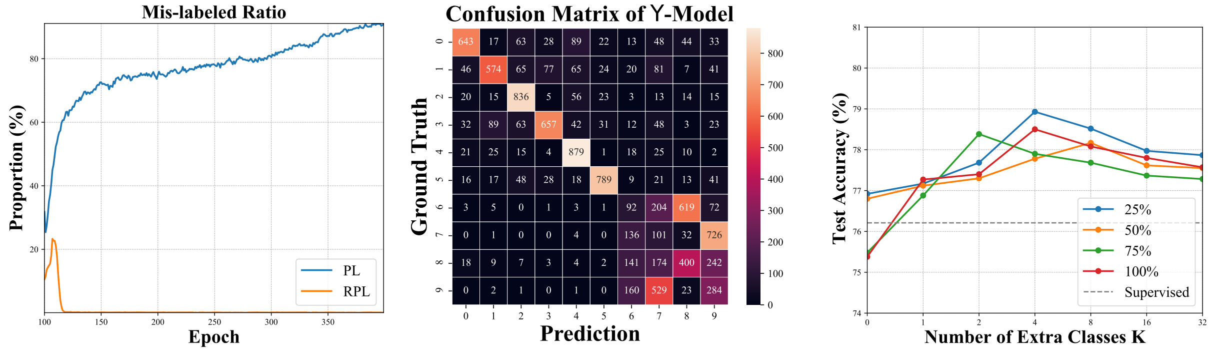
Table 2: Comparison of three mid/large-scale datasets with different class-mismatched ratios. The backbone is Wide-ResNet-28-2 for all experiments. The standard deviations of the accuracies over five runs are also reported. The best results are highlighted in **bold**. The second-best results are highlighted in underline. Baseline and Oracle have the same meaning as in Section 3.2.

Mismatch Ratio	0%	25%	50%	75%	100%
			CIFAR100 (50/50)		
PL	61.68 ± 0.30	60.20 ± 0.25	60.12 ± 0.30	57.79 ± 0.27	57.62 ± 0.57
UASD	60.30 ± 0.26	59.42 ± 0.61	59.92 ± 0.53	58.94 ± 0.63	58.74 ± 0.50
DS^{3}L	60.68 ± 0.67	60.60 ± 0.40	59.22 ± 0.33	59.74 ± 0.37	**59.56 ± 0.57**
Υ-Model	**62.10 ± 0.30**	**61.26 ± 0.40**	60.68 ± 0.24	60.12 ± 0.38	59.46 ± 0.36
ORCA	58.94 ± 0.24	59.98 ± 0.35	60.14 ± 0.40	58.84 ± 0.27	44.00 ± 0.42
OpenMatch	62.08 ± 0.26	60.94 ± 0.36	**60.92 ± 0.23**	**60.36 ± 0.47**	59.22 ± 0.50
Baseline			58.68 ± 0.25		
Oracle	75.46 ± 0.20	73.36 ± 0.29	72.94 ± 0.25	69.26 ± 0.26	63.90 ± 0.30
			Tiny ImageNet (100/100)		
PL	43.42 ± 1.03	42.88 ± 1.51	41.94 ± 1.49	39.72 ± 2.30	38.94 ± 2.41
UASD	43.34 ± 0.78	42.34 ± 0.61	41.80 ± 1.33	41.08 ± 1.16	36.16 ± 1.05
DS^{3}L*	-	-	-	-	-
Υ-Model	**44.42 ± 0.43**	**43.48 ± 0.40**	**42.42 ± 0.95**	**43.22 ± 0.60**	**41.76 ± 0.63**
ORCA	42.16 ± 0.41	40.22 ± 0.37	41.58 ± 1.24	42.92 ± 0.57	40.82 ± 0.36
OpenMatch	43.42 ± 0.31	42.00 ± 0.37	42.26 ± 0.27	42.5 ± 0.31	41.03 ± 0.38
Baseline			39.66 ± 0.52		
Oracle	56.18 ± 0.31	53.8 ± 0.28	51.82 ± 0.20	48.3 ± 0.30	45.28 ± 0.31
			ImageNet (50/50)		
PL	50.04 ± 0.11	49.32 ± 0.15	48.36 ± 0.19	47.40 ± 0.13	46.96 ± 0.19
UASD	50.24 ± 0.20	48.68 ± 0.26	48.12 ± 0.17	48.00 ± 0.11	47.04 ± 0.25
DS3L	-	-	-	-	-
Υ-Model	**50.6 ± 0.19**	49.78 ± 0.16	**48.88 ± 0.30**	**48.36 ± 0.11**	**47.64 ± 0.22**
ORCA	41.56 ± 0.27	43.32 ± 0.38	46.60 ± 0.23	47.08 ± 0.45	46.88 ± 0.36
OpenMatch	49.92 ± 0.11	**49.88 ± 0.15**	48.82 ± 0.19	48.01 ± 0.21	46.84 ± 0.27
Baseline			48.12 ± 0.29		
Oracle	62.92 ± 0.26	62.00 ± 0.28	61.04 ± 0.26	57.92 ± 0.26	55.96 ± 0.30

* We cannot finish DS^{3}L on mid-scale or large-scale datasets like Tiny ImageNet or ImageNet100 within a reasonable time.

from OOD classes. This is attributed to the fact that DS^{3}L uses a down weighting strategy to alleviate the negative effect of OOD data and does not change the form of unsupervised loss. But we have shown in Section 3.2 that labeling OOD data as ID classes damages performance anyhow. On the contrary, Υ-Model uses the OOD data in the right way – simulating the process of training them with their ground truth labels. As a result, our method shows superiority especially under a large class-mismatched ratio. We also notice that the performance curve of Υ-Model appears a U-shape (obvious on CIFAR10). A possible reason is that RPL and SEC compete with each other. RPL tends to make samples get a high prediction on ID classes while SEC tends to make samples get a high prediction on OOD classes. When the class-mismatched ratio reaches 0% (100%), RPL (SEC) dominates the other. In this circumstance, one works without any disturbance to the other. However, when the class-mismatched ratio is 50%, they compete fiercely with each other, causing many incorrectly recognized ID or OOD samples.

Additionally, we compare our method to OpenMatch (Saito et al., 2021) and ORCA (Cao et al., 2021) on mid-/large-scale datasets in Section 6. *Again we use the same augmentation for both labeled and unlabeled data to avoid unfairness for baseline.* ORCA is an open-world semi-supervised learning method, which not only aims to correctly classify ID data but also to cluster OOD. Due to this, its ID classification performance is lower than others. OpenMatch is a competitive method, but it can not surpass ours in most of the settings.



(a) (b) (c)

Figure 7: Two experiments on CIFAR10 benchmark to validate the functionality of RPL and SEC. (a) The proportion of OOD data pseudo-labeled as ID classes with the training goes on. Vanilla PL makes this ratio keep increasing while RPL makes it drop to 0. It demonstrates that RPL help filter out OOD data. (b) Confusion matrix on the full test set of CIFAR10. RPL solves the imbalance problem.

6.3 Ablation Study

In this section, we validate the functionality of RPL and SEC. We conduct experiments on CIFAR10 benchmark as in the analysis section 3.

Table 3: Validation of RPL and SEC. The experiments are conducted on CIFAR10 with a Wide-ResNet-28-2 backbone. Class-mismatched ratio varies from 0 to 100%. Check RPL or not means using RPL or vanilla PL. $K = 0$ means we do not use SEC. $K = 1$ means we label all the low confidence data as a unified class $K_{ID} + 1$.

RPL	SEC	K	Class-Mismatched Ratio (%)				
			0	25	50	75	100
		0	78.23 ± 0.39	76.50 ± 0.30	75.85 ± 0.35	74.60 ± 0.27	73.97 ± 0.30
✓		0	**78.76 ± 0.49**	76.92 ± 0.32	76.80 ± 0.31	75.48 ± 0.27	75.38 ± 0.27
✓	✓	1	**78.86 ± 0.37**	77.17 ± 0.55	77.12 ± 0.37	76.88 ± 0.35	77.27 ± 0.39
	✓	4	78.11 ± 0.45	77.43 ± 0.29	77.46 ± 0.37	76.38 ± 0.37	74.18 ± 0.32
✓	✓	4	**78.98 ± 0.49**	**78.93 ± 0.28**	**77.78 ± 0.31**	**77.90 ± 0.21**	**78.50 ± 0.27**
Baseline			76.21 ± 0.21				

Validation of effectiveness of RPL and SEC. We conduct ablation studies under different class-mismatched ratios and report the averaged test accuracy and standard deviation of five runs. As usual, we vary the class-mismatched ratio. Table 3 displays the results. Firstly, comparing the first line and second line of the table, RPL not only outperforms vanilla PL in high class-mismatched ratio scenarios but also improves in low class-mismatched ratio scenarios. This reveals that balanced pseudo-labels always help since once the model creates imbalanced pseudo-labels, it will deteriorate when there are not enough measures to correct it. Secondly, comparing the second and third line, it shows that RPL alone alleviate the performance degradation but it can not prevent it, in accord with Conclusion 3. When using SEC, ϒ-Model gets better results than supervised baseline when the class-mismatched ratio is high. Besides, comparing the third and last lines, we see that when clustering OOD data and exploring their semantics instead of using a unified class to label them, the performance improves.

RPL helps filter out OOD data and solve the imbalance problem. It is noticeable that the last two lines of Table 3 show that without RPL, SEC alone can not achieve better performance than supervised baseline. We show the reason here. Figure 7(a) plots the proportion of OOD data that are pseudo-labeled as ID classes. Without RPL, *i.e*, using vanilla PL, the number of incorrectly recognized OOD data keep increasing as the training proceeds. While with RPL, this ratio rapidly drops to 0. This proves that RPL help filter out OOD data by utilizing the imbalance property of OOD data. Further, we present the confusion matrix of Υ-Model on the full test set (all the 10 classes) of CIFAR10. Compared to vanilla PL in Figure 3(b), Υ-Model does not suffer from imbalance problem, as a result of which, its performance is not degraded.

Effect of extra class number K. We vary the number of extra classes K. Figure 7(c) shows the result on CIFAR10 with various class mismatch ratios. Without SEC ($K = 0$), Υ-Model underperforms the supervised baseline. Using SEC ($K \geq 1$), Υ-Model is always better than baseline. Also, it reaches its best performance when K is roughly the actual number of OOD classes. This demonstrates that by simulating the process of training on OOD data with ground truth, SEC helps the classification model on ID data. Also, our model benefits from SEC but is not sensitive to the selection of K. SEC helps explore the semantic structure of OOD data, but it does not affect the learning on ID data since they perform tasks on different classifiers.

Table 4: Comparison of PL and Υ-Model on the constructed class-mismatched CIFAR10 dataset with imbalanced OOD data.

Class-Mismatched Ratio (%)	25	50	75	100
PL	76.42	75.78	74.58	73.33
Υ-Model (Ours)	**77.57**	**77.42**	**77.13**	**77.26**

Effectiveness on imbalanced OOD data. It might be a question that whether it is suitable to use balanced clustering since the OOD data are usually imbalanced in the real world. We note that we use cluster methods only to mine the semantics among OOD data. We do not aim to correctly recognize them. The effectiveness of using clustering methods to learn semantically meaningful representations has been proved in many works (Asano et al., 2020; Caron et al., 2020). To prove the effectiveness of our method, we conduct experiments on an additional benchmark *where the OOD data is imbalanced*. The OOD data are subsampled like CIFAR-10-LT (Cao et al., 2019) with imbalance ratio of 10 on the OOD classes. Table 4 displays the results. We can see that our model can handle situations where OOD data are even imbalanced.

Other imbalance methods and uncertainty measures. We present results of an imbalanced method applied to PL : PL-Reweight and two uncertainty measures applied to our model: Υ-Model (Ent) and Υ-Model (SD). Ent and SD detect OOD data via entropy and score difference. Ent uses negative entropy of the output distribution as the confident measure. SD use the difference between the largest and the second-largest output probability as the confidence score.

Table 5: Experiments on combining different imbalance and OOD detection methods with PL/Υ-Model. PL-Reweight uses reweight strategy to balance pseudo-labels. Ent (Entropy) uses negative entropy of the output distribution as the confident measure. SD (Score Difference) uses the difference between the largest and the second-largest output probability as the confidence score.

Class-Mismatch Ratio (%)	25	50	75	100
PL	76.5	75.85	74.6	73.97
PL-Reweight	76.98	76.33	74.52	diverge
Υ-Model (Ent)	77.78	77.97	76.32	77.12
Υ-Model (SD)	77.70	**78.25**	**78.02**	76.72
Υ-Model (Ours)	**78.93**	77.78	77.90	**78.50**

Common imbalance methods like resample and reweight will be *unstable* in this setting since there is the possibility that the number of pseudo-labels is zero in certain classes. Also, in Section 3.3, we have concluded that assigning ID labels for OOD data will degrade the model. Therefore, solving the imbalance problem is only a partial solution for this setting. In contrast, our RPL is stable in this situation and uses the imbalance phenomenon to simultaneously filter out OOD data. Also, uncertainty measures other than confidence score are applicable to our method.

7 Conclusion

In this paper, we analyze Pseudo-Labeling in class-mismatched semi-supervised learning where there are unlabeled OOD data from other classes. We show that Pseudo-Labeling suffers from performance degradation due to imbalanced pseudo-labels on OOD data. The correct way to use OOD data is to label them as classes different from ID classes while also partitioning them according to their semantics. Based on the analysis, we proposed Υ-Model and empirically validate its effectiveness.

In future work, we will explore whether other forms of semi-supervised learning methods like the consistency-based method suffer from the same problems. Also, currently our model is based on Pseudo-labeling. But we see the possibility of extending it to other SSL methods. For example, there may be re-balanced consistency methods and SEC may be a plug-and-play component for all SSL methods.

8 Acknowledgments

This work is partially supported by The National Key R&D Program of China (2020AAA0109401), NSFC (62006112, 61773198, 61921006), NSF of Jiangsu Province (BK20200313), CCF-Baidu Open Fund (NO.2021PP15002000).

References

Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *IJCNN*, pp. 1–8. IEEE, 2020.

Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2020.

Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. There are many consistent explanations of unlabeled data: Why you should average. In *ICLR*, 2019.

Abhijit Bendale and Terrance E. Boult. Towards open set deep networks. In *CVPR*, pp. 1563–1572, 2016.

David Berthelot, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *NeurIPS*, pp. 5050–5060, 2019.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Aréchiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *NeurIPS*, pp. 1565–1576, 2019.

Kaidi Cao, Maria Brbic, and Jure Leskovec. Open-world semi-supervised learning. *CoRR*, abs/2102.03526, 2021.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, pp. 139–156, 2018.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020.

Yanbei Chen, Xiatian Zhu, Wei Li, and Shaogang Gong. Semi-supervised learning under class distribution mismatch. In *AAAI*, pp. 3569–3576, 2020.

Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In *NeurIPS*, 2020.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NIPS*, pp. 2292–2300, 2013.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pp. 248–255, 2009.

Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. SCAN: learning to classify images without labels. In *ECCV*, pp. 268–285, 2020.

Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *NIPS*, pp. 529–536, 2004.

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 *ICML*, pp. 3897–3906, 2020.

Kai Han, Andrea Vedaldi, and Andrew Zisserman. Learning to discover novel visual categories via deep transfer clustering. In *ICCV*, pp. 8400–8408, 2019.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. Revisiting self-training for neural sequence generation. In *ICLR*, 2020.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017.

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 *ICCV*, pp. 8290–8299, 2021.

Zhuo Huang, Ying Tai, Chengjie Wang, Jian Yang, and Chen Gong. They are not completely useless: Towards recycling transferable unlabeled data for class-mismatched semi-supervised learning. *CoRR*, abs/2011.13529, 2020.

Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *CVPR*, pp. 5070–5079, 2019.

Diederik P. Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *NIPS*, pp. 3581–3589, 2014.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017.

Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, pp. 896, 2013.

Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pp. 2999–3007, 2017.

Huixiang Luo, Hao Cheng, Yuting Gao, Ke Li, Mengdan Zhang, Fanxu Meng, Xiaowei Guo, Feiyue Huang, and Xing Sun. On the consistency training for open-set semi-supervised learning. *CoRR*, abs/2101.08237, 2021.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *TPAMI*, 41(8):1979–1993, 2019.

Avital Oliver, Augustus Odena, Colin Raffel, Ekin Dogus Cubuk, and Ian J. Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *NeurIPS*, pp. 3239–3250, 2018.

Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. In *WACV*, pp. 29–36, 2005.

Kuniaki Saito, Donghyun Kim, and Kate Saenko. Openmatch: Open-set semi-supervised learning with open-set consistency regularization. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *NeurIPS*, pp. 25956–25967, 2021.

Walter J. Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E. Boult. Toward open set recognition. *TPAMI*, 35(7):1757–1772, 2013.

Weiwei Shi, Yihong Gong, Chris Ding, Zhiheng Ma, Xiaoyu Tao, and Nanning Zheng. Transductive semi-supervised deep learning using min-max features. In *ECCV*, pp. 311–327, 2018.

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 *NeurIPS*, 2020.

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

Qizhe Xie, Minh-Thang Luong, Eduard H. Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification. In *CVPR*, pp. 10684–10695, 2020.

Bo Yang, Xiao Fu, Nicholas D. Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *ICML*, pp. 3861–3870, 2017.

Qing Yu, Daiki Ikami, Go Irie, and Kiyoharu Aizawa. Multi-task curriculum framework for open-set semi-supervised learning. In *ECCV*, pp. 438–454, 2020.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016.

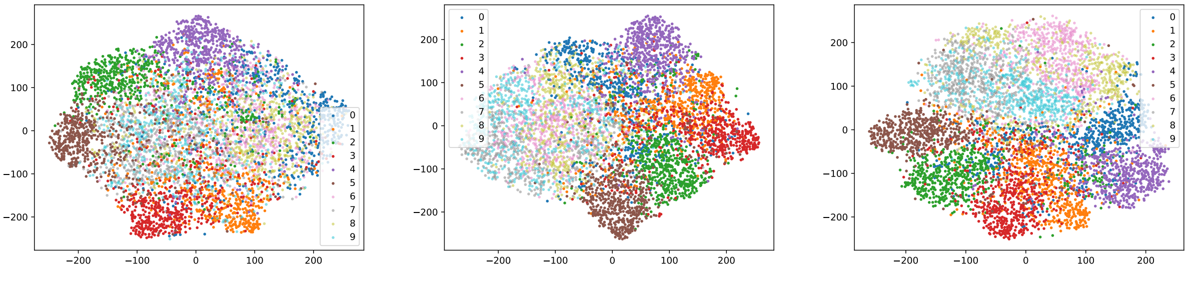
A Algorithm

Algorithm 1 Υ-Model algorithm

```
Require: Labeled dataset $\mathcal{D}_l = \{(\boldsymbol{x}_{li}, y_{li})\}_{i=1}^{n}$, and unlabeled dataset $\mathcal{D}_u = \{\boldsymbol{x}_{ui}\}_{i=1}^{m}$; Classification model $f_\phi$ parameterized with $\phi$, ID class number $K_{ID}$, extra class number $K$, total epoch number $E$, pretrain epochs $E_{pt}$, interval to update pseudo-labels $E_{pl}$, pseudo-labeled set $\mathcal{P}$, confidence calculation function $c$.
function RebalancedPseudoLabeling($\mathcal{D}, f, \tau$)
    $N \leftarrow \min_{y \in \mathcal{Y}_{ID}} |\{\boldsymbol{x} \in \mathcal{D}_u \mid f(y \mid \boldsymbol{x}) > \tau\}|$
    $\tau_y \leftarrow Nth\_biggest(\{f(y \mid \boldsymbol{x}) \mid \boldsymbol{x} \in \mathcal{D}_u\}), \quad y = 1, 2, \ldots, K_{ID}$
    $\mathcal{P} \leftarrow \bigcup_{y \in \mathcal{Y}_{ID}} \{(\boldsymbol{x}, y) \mid f(y \mid \boldsymbol{x}) \geq \tau_y, \boldsymbol{x} \in \mathcal{D}_u\}$
    return $\mathcal{P}$
function SemanticExplorationClustering($\mathcal{D}, f, \gamma$)
    $S \leftarrow \{\boldsymbol{x} | c(x) < \gamma\}$
    $M \leftarrow |S|$
    $P_{ij} \leftarrow f(K_{ID} + i|\boldsymbol{x}_j) / \sum_{k=1}^{K} f(K_{ID} + k|\boldsymbol{x}_j), \quad i = 1, 2, \ldots, K, \quad j = 1, 2, \ldots, M$
    Solve 7 by Sinkhorn-Knopp algorithm and get $Q$
    $\hat{y}_j \leftarrow K_{ID} + \arg\max_i Q_{ij}, \quad j = 1, 2, \ldots, M$
    $\mathcal{C} \leftarrow \{(\boldsymbol{x}_j, \hat{y}_j)\}_{j=1}^{M}$
    return $\mathcal{C}$
for e = 1 to $E$ do
    if e $< E_{pt}$ then
        train $f_\phi$ with standard supervised learning on $\mathcal{D}_l$          ▷ Pre-training phase
    else
        train $f_\phi$ with standard supervised learning on $\mathcal{D}_l \cup \mathcal{P}$          ▷ PL training phase
    if e $\leq E_{pt}$ and e $\% E_{pl} = 0$ then
        $\mathcal{P} \leftarrow \emptyset$
        $\mathcal{P}_\tau \leftarrow$ RebalancedPseudoLabeling($\mathcal{D}_u, f_\phi, \tau$)          ▷ Perform RPL
        $\mathcal{P}_\gamma \leftarrow$ SemanticExplorationClustering($\mathcal{D}_u, f_\phi, \gamma$)          ▷ Perform SEC
        $\mathcal{P} \leftarrow \mathcal{P}_\tau \cup \mathcal{P}_\gamma$
return classification model $f_\phi$
```

B Embedding visualization

We visualize the embedding of supervised baseline, vanilla PL and υ-Model on CIFAR10's test set with all the classes by t-SNE. Figure 8 shows the result. OOD data are mixed with ID data since the supervised baseline does not see unlabeled OOD data. PL mixes OOD data with samples of certain classes (class 0). This is attributed to their pseudo-labels being biased toward this class. Also, we can not clearly distinguish between OOD classes. In contrast, Υ-Model can not only make ID data distinguishable but also forms meaningful clusters on OOD data.



(a) Baseline

(b) PL

(c) Υ-Model

Figure 8: t-SNE visualization of supervised baseline, vanilla PL and Υ-Model on CIFAR10's test set. Class 0-5 are ID classes, shown by non-transparent circles. Class 6-9 are OOD classes, represented by semi-transparent circles.

C Imbalance of Pseudo-Labels

C.1 Metrics

To demonstrate the imbalance of pseudo-labels on OOD data, we compute two metrics to measure the extent of imbalance.

- The KL divergence of pseudo-label distribution q and the uniform distribution u:

$$kl = KL(q||u) = \sum_i q_i \log \frac{q_i}{u_i}$$

- The ratio of the 'majority class' and 'minority class':

$$r = \frac{\max_i q_i}{\min_i q_i}$$

The results on these datasets are displayed in Table 6.

Table 6: Illustration of the extent of imbalance of pseudo-labels on different datasets.

	C10 (6/4)	**C10 (5/5)**	**SVHN (6/4)**	**C100 (50/50)**	**TIN (100/100)**
			kl		
ID	0.0131	0.0169	0.0006	0.0771	0.0728
OOD	0.3636	0.0429	0.1031	0.4177	0.4089
			r		
ID	1.5390	1.6235	1.1047	5.1834	3.6609
OOD	18.3183	2.4878	4.4797	213.7661	187.2124

C.2 Different Selected OOD Classes

In this section, we show that the imbalance phenomenon is general in *real-world datasets*. We conduct experiments on CIFAR10(6/4), SVHN(6/4), CIFAR100(50/50) and ImageNet100(50/50). The three datasets cover both small-scale and large-scale datasets, cover datasets with both small and large number of classes, and also cover datasets with hierarchical (CIFAR100) and non-hierarchical (SVHN) classes. We also add a dataset with 6 random ID classes from CIFAR10 and 4 random OOD classes from SVHN, denoted as CIFAR10(6)/SVHN(4). This mixed dataset simulates the condition where all the OOD data come from domain largely different from the ID data. For each dataset, we randomly select different set of classes as the ID classes, and the remaining classes as OOD classes except CIFAR10(6)/SVHN(4). A classification model is trained on ID classes and the imbalance ratio on both ID and OOD classes is reported. We repeat for 50 times each dataset.

Figure 9 display the results on these datasets. We can draw some conclusions from this figure. First, in each trial, the imbalance ratio on ID classes is always lower than OOD by a large margin. It proves that the imbalanced pseudo labels are common on OOD data. Second, imbalance ratios have much lower deviation on ID classes than on OOD classes. It is natural since ID data are selected from the same distribution as the training data, while the OOD data come from a irrelevant domain. Therefore, pseudo-labels on these OOD data appear with high randomization, which means high imbalance ratio and high deviation.

Discussion. We agree that whether pseudo-labels on OOD data are imbalanced depends on how they are generated. It is always possible to manually pick a set of OOD data that have balanced pseudo-labels. However, we are here to show that for a *non-curated natural dataset*, the imbalance on OOD data is a general phenomenon. With little possibility, the pseudo-labels happen to be balanced on these data. As a result of it, our method is effective in most conditions.

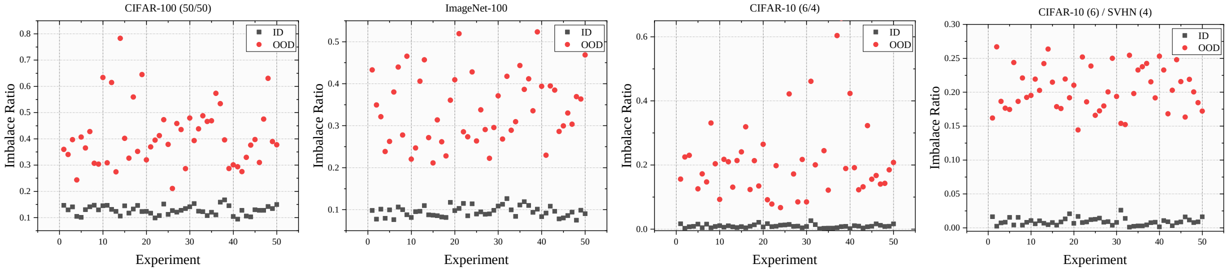


Figure 9: Imbalance ratio on ID and OOD classes with different generations of ID and OOD data. It can be seen that pseudo-labels on ID classes always are always (relatively) balanced. The imbalanced ratios of ID classes concentrate around a certain low value. In contrast, pseudo-labels on OOD data have a much higher imbalanced ratio. Among different OOD settings, the imbalance ratios vary much but all of them lie higher than ratio of ID classes by a large margin.

D Comparison with FixMatch and MixMatch

We display the results of FixMatch (Sohn et al., 2020) and MixMatch (Berthelot et al., 2019). For a fair comparison, we use the same augmentation for labeled and unlabeled data here. The augmentation 'paper' means what we used in the paper. It is commonly used in class-mismatched settings (Oliver et al., 2018; Guo et al., 2020). RandAug (Cubuk et al., 2020) is the augmentation used in FixMatch.

Table 7: Results of FixMatch, MixMatch, Pseudo-Labeling (PL) and our method. The results are reported on CIFAR-10 (6/4) with different class mismatch ratios.

	Augmentation	0	25%	50%	75%	100%
MixMatch	paper	77.88	77.15	76.83	75.9	75.85
PL	paper	78.23	76.5	75.85	74.6	73.97
FixMatch	paper	78.75	76.25	diverge	diverge	diverge
Υ-Model (ours)	paper	**78.98**	**78.93**	**77.78**	**77.9**	**78.5**
Baseline	paper			76.21		
MixMatch	RandAug	81.98	80.08	76.75	73.45	69.33
PL	RandAug	88.30	86.80	85.61	83.72	81.62
FixMatch	RandAug	87.35	84.314	82.367	78.733	diverge
Υ-Model (ours)	RandAug	**88.35**	**86.83**	**85.91**	**85.25**	**84.31**
Baseline	RandAug			83.75		

Some conclusions can be drawn from Table 7: FixMatch is unstable in the class-mismatched setting while MixMatch is more stable. The former is mainly caused by the imbalance of pseudo-labels. MixMatch's stability may be brought about by its mixup operation. FixMatch and MixMatch all suffer from performance degradation in such a setting. The strong augmentation strategy in FixMatch can bring improvement to our method. It brings improvement to all the methods, but we also emphasize that it also pulls up the baseline.

E Hyperparameters on Different Datasets

We compare our method with vanilla PL and the two class-mismatched methods in Section 6.2. We use the following hyperparameters:

- **CIFAR10 (6/4)**: $\tau = 0.95, \gamma = 0.3, E_{pt} = 50, E_{pl} = 2, K = 4$
- **SVHN (6/4)**: $\tau = 0.95, \gamma = 0.3, E_{pt} = 50, E_{pl} = 2, K = 4$

- **CIFAR100 (50/50)**:$\tau = 0.95, \gamma = 0.18, E_{pt} = 50, E_{pl} = 2, K = 20$
- **Tiny ImageNet (100/100)**: $\tau = 0.9, \gamma = 0.15, E_{pt} = 50, E_{pl} = 2, K = 20$
- **ImageNet (50/50)**: $\tau = 0.9, \gamma = 0.20, E_{pt} = 50, E_{pl} = 2, K = 20$

For **CIFAR100 (50/50)** , **Tiny ImageNet (100/100)** and **ImageNet (50/50)**, we use a weight factor λ to trade off the loss on labeled set $\mathcal{D}_l$ and pseudo-labeled set $\mathcal{P}$, which ramps up with function $\lambda = \exp\left(-5 \times \left(1 - \min\left(\frac{iter}{40{,}000}, 1\right)\right)^2\right)$, where $iter$ is the number of training steps from E_{pt}.

F Analysis of Time Complexity

It may be the concern that introducing an extra clustering component causes the Υ-Model to be too computationally expensive to be practical. We want to emphasize that it is not the case. We have empirically tested the run time Υ-Model compared with vanilla Pseudo-Labeling. We display the comparison results in Figure 10. The upper two figures show the test accuracy and relative runtime (compare to vanilla Pseudo-Labeling) with varying numbers of extra classes K. It can be shown that the time complexity is almost irrelevant to K. The lower two figures show the relationship to the Sinkhorn iterations, which control the quality of clustering (Cuturi, 2013; Asano et al., 2020). It can be seen that both the performance and time complexity increases with the number of iterations. However, we want to note that the X-axis is in log scale. When the number increases exponentially, the time increases nearly linearly. Even though we use 32 iterations, which is 6x the number in Asano et al. (2020), it cost not more than 1.7x time of vanilla Pseudo-Labeling.

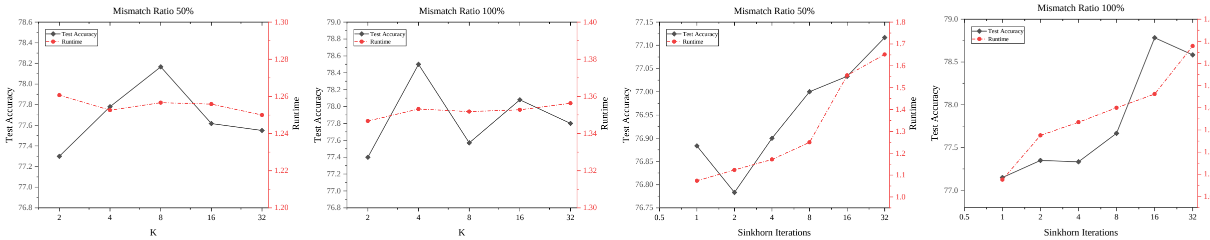


Figure 10: The test accuracy and relative runtime (compare to vanilla Pseudo-Labeling) v.s. the number of clusters K and Sinkhorn iteration. The time complexity is not relevant to K, but it affects the test accuracy. Both the accuracy and time complexity increase with the number of iterations. But when the number of iterations increases exponentially, the time increases nearly linearly.

Our method is not inefficient as it may seem. First, the OT-based clustering is fast by the Sinkhorn-Knopp algorithm (Cuturi, 2013). Several works using clustering have been proved efficient even on large-scale datasets (Asano et al., 2020; Caron et al., 2020). Our SEC component has similar functionality and setup in practice. Second, we perform clustering periodically. We do clustering every 2 epochs. Compared to the network updating cost, clustering spends an acceptable time.

G Discussion about Limitation and Social Impacts

Currently, our model is based on Pseudo-labeling. In future work, we will explore whether other forms of semi-supervised learning methods like the consistency-based method suffer from the same problems. And we will investigate whether the rebalance and semantic explore strategy can also benefit other forms of semi-supervised learning methods.

This paper explores a way of saving Pseudo-labeling methods from the harm of class-mismatched unlabeled data. The results of this paper can spare the effort of cleaning unlabeled data, which can benefit the development and application of semi-supervised learning. Currently, we do not see a direct negative impact on society.