The Hebrew University of Jerusalem
The Rachel and Selim Benin School of Computer Science and Engineering

Semi-Supervised Learning in the Few-Shot Zero-Shot Scenario

למידה סמי־מפוקחת עם מעט תיוגים כאשר קיימות מחלקות לא מתוייגות

Noam Shimon Fluss

Thesis submitted in partial fulfillment of the requirements
for the Master of Sciences degree
in Computer Science

Under the supervision of **Prof. Daphna Weinshall**

October 2023

האוניברסיטה העברית בירושלים
בית הספר להנדסה ולמדעי המחשב על שם רחל וסלים בנין
החוג למדעי המחשב

Semi-Supervised Learning in the Few-Shot Zero-Shot Scenario

למידה סמי־מפוקחת עם מעט תיוגים כאשר קיימות מחלקות לא מתוייגות

מוגש על ידי
נועם שמעון פלוס

עבודת גמר לתואר מוסמך במדעי המחשב

עבודה זו הונחתה על ידי
פרופ' דפנה וינשל

חשון תשפ"ד

תקציר

למידה סמי־מפוקחת מנצלת את המידע המתויג והמידע הלא מתויג על מנת לשפר את הביצועים. בדרך כלל אחת מההנחות על המידע המתויג והמידע הלא מתויג היא שהם חולקים את אותם מחלקות. אף על פי כן, בעולם האמיתי, במיוחד כאשר כמות המידע המתויג קטנה, יש סוגי מחלקות בלי דוגמאות מתויגות הנראים באימון. אלגוריתמים קיימים מנסים לדחות את כל ההמחלקות בלי דוגמאות מתויגות באימון OPEN SET SSL או הפרדה בין מחלקות בלי דוגמאות תמויגות רק על המידע הלא מתויג באימון OPEN WORLD SSL בעבודה זו אנחנו בונים מסווג לדוגמאות מהמחלקות עם ובלי דוגמאות מתויגות באימון. הגישה שלנו מבוססת על שדרוג אלגוריתם למידה סמי־מפוקחת, כמו FLEXMATCH, על ידי הוספה של שגיאת אנטרופיה במהלך האימון. תוספת זו מאפשר לשדרג את האלגוריתמים הקיימים להיות מסוגלים לסווג מחלקות נראות ולא נראות. אנחנו מדגימים את השיפור המשמעותי שאנחנו מביאים אל מול אלגוריתמים עדכניים של למידה סמי־מפוקחת, OPEN SET SSL ו OPEN WORLD SSL על גבי שני סוגי מאגר תמונות 100 CIFAR ו 10 STL. השיפור משמעותי ביותר כאשר יש כמות מוגבלת של מידע מתויג באימון (1 - 25 דוגמאות מתויגות למחלקה).

Abstract

Semi-Supervised Learning (SSL) leverages both labeled and unlabeled data to improve model performance. Traditional SSL methods assume that labeled and unlabeled data share the same label space. However, in real-world applications, especially when the labeled training set is small, there may be classes that are missing from the labeled set. Existing frameworks aim to either reject all unseen classes (open-set SSL) or to discover unseen classes by partitioning an unlabeled set during training (open-world SSL). In our work, we construct a classifier for points from both seen and unseen classes. Our approach is based on extending an existing SSL method, such as *FlexMatch*, by incorporating an additional entropy loss. This enhancement allows our method to improve the performance of any existing SSL method in the classification of both seen and unseen classes. We demonstrate large improvement gains over state-of-the-art SSL, open-set SSL, and open-world SSL methods, on two benchmark image classification data sets, CIFAR-100 and STL-10. The gains are most pronounced when the labeled data is severely limited (1-25 labeled examples per class).

Acknowledgements

I would first like to thank my supervisor, Prof. Daphna Weinshall, for helping, guiding and encouraging me during the research. I would also like to thank Guy Hacohen for researching and assisting me with countless technical problems along the way. Finally I thank all of my lab members that were there during my research.

Contents

1 Introduction **1**

1.1 Our proposed approach . . . 3

2 Previous Work **4**

2.1 Closed and Open Semi-Supervised Learning . . . 4

2.2 Open-World SSL . . . 6

2.3 Summary of contribution . . . 7

3 Methods **9**

3.1 Notations and definitions . . . 9

3.2 Evaluation scores . . . 10

3.3 Our Method . . . 11

3.4 Baseline methods for comparison . . . 12

4 Empirical Results **14**

4.1 Methodology and technical details . . . 14

4.2 Main Results . . . 15

4.3 Ablation Study . . . 18

5 Conclusion **20**

List of Figures

3.1 The combined SSL loss used in our approach. 11

4.1 CIFAR-100: balanced labeled set and few-shot for seen classes 16

4.2 CIFAR-100: non-uniform labeled set 17

4.3 CIFAR-100: balanced labeled set and many-shot for seen classes 17

4.4 STL-10: balanced labeled set and few-shot for seen classes 18

4.5 Comparison of Combined Score with and without K-means Adaptation . . . 18

List of Tables

2.1 Overview of related frameworks 4

4.1 Overview of convergence time 19

1 Introduction

As the collection of data becomes increasingly widespread, new scenarios that pose challenges to the traditional supervised machine learning framework are continuously emerging. Consider, for instance, a scenario where data can be easily collected at a low cost. However, acquiring accurate labels for this data requires specialized expertise or expensive machinery, all while adhering to strict budget constraints. Such scenarios may occur more frequently in fields like medical research or genomics. In this scenario, the unlabeled dataset comprises a vast amount of data, but due to limited resources, only a small portion of it can be selected in an unsupervised manner to be further processed for labeling[1]. In the end, we are left with a substantial unlabeled dataset and a relatively limited number of labeled data points.

In this scenario, we encounter a unique challenge where certain classes are missing from the labeled set, to be denoted "zero-shot" or "unseen" classes. Note also that the small labeled set is unlikely to accurately represent the prior probability distribution of the classes. These remaining classes are called "imbalanced few-shot" or "seen". As a result, in the proposed scenario, the labeled set is constructed of classes with only few-shot and zero-shot examples. Additionally, due to the random selection of points to be processed for labeling, the class distribution of the labeled points may deviate widely from the true class distribution.

Nevertheless, we identify valuable constraints in this scenario, that can be leveraged for the successful handling of zero-shot classes. Specifically, by assumption the unlabeled training set is derived from the same distribution as the test dataset, which consists of new samples obtained under similar conditions. Thus there are no Out-Of-Distribution (OOD) samples to consider, Additionally, the set of possible labels is fixed throughout.

While Semi-Supervised Learning (SSL) appears to be a suitable framework to address the

[1]We note that active learning in the low budget domain aims to replace the random selection of the set of points to be labeled by a more effective selection methodology.

problem at hand, it is important to note that most SSL methods assume the presence of all classes in the labeled set, namely, that there are no zero-shot classes. A step in the right direction is taken in the *open-set SSL* (OSSL) framework, where the data contains points sampled from either unseen classes or OOD (or both). However, open-set techniques are designed to reject test samples that do not belong to any seen class and do not predict missing class labels. Nevertheless, we compare our proposed method to state-of-the-art SSL and OSSL methods adapted to identify unseen classes (see details in Section 3.4), showing large gains in performance for our approach.

A closely related scenario is addressed in the *open-world SSL* framework, which aims to classify seen classes **and** to discover unseen classes in an unlabeled set. Typically in this framework, a learner is given substantial labeled and unlabeled sets for training, where both sets are sampled from the same distribution. Originally, the aim of such methods was to partition the unlabeled set in a transductive manner. Such methods can be extended to our inductive scenario by providing the partitioned unlabeled set as input to a regular SSL method, see discussion of related work below.

In our work we address the inductive problem of **unseen class classification**. It is important to appreciate the difference between unseen class **discovery** and unseen class **classification**. In discovery, the obtained structure is not universal, and only applies to the given unlabeled set. Consequently, when given two different test sets possibly at two different time points, the whole pipeline needs to be run again, and the outcome may be inconsistent in the sense that different data partitions are not always easy to consolidate into a single coherent partition.

In contrast, our method provides a universal partition of the data, and a classifier that can identify the cluster identity of each future test-point in a consistent manner and irrespective of other test points. The only uncertainty remains in the naming of clusters: if our unseen classes are dogs and cats, for example, a classifier of unseen examples may identify dogs as cluster 1, and cats as cluster 2. Without labels it cannot decide which cluster to call dogs and which one cats. Nevertheless, if the algorithm is successful, from now on all future images of dogs will be assigned to cluster 1, and images of cats to cluster 2.

Another related methodology is Zero-Shot Learning (ZSL). Traditionally, ZSL methods are designed to train models that can classify objects from unseen classes by utilizing semantic side-information and knowledge acquired from seen classes [see survey by 15]. Differently, in our scenario, we lack access to any supplementary semantic information. Another difficulty with adopting this framework to our scenario stems from the fact that ZSL usually assumes that seen classes are adequately sampled, i.e., they are **not** "few-shot" but "many-shot".

1.1 Our proposed approach

We describe a simple end-to-end framework that tackles the above scenario, offering a general solution that can be used to enhance any semi-supervised learning (SSL) method. The approach consists of two main steps. (i) Modify the last layer of a given SSL architecture to incorporate both seen and unseen classes. By doing so, the model is capable of handling and classifying instances from both types of classes. (ii) Introduce an additional term into the loss function, which penalizes discrepancies between the output distribution of the trained model and the expected class distribution.

As the field of semi-supervised learning continues to evolve, the proposed approach can be used to adapt future SSL methods to our scenario. When compared to zero-shot learning, instead of relying on hard-to-collect semantic side information, we exploit the fact that the unlabeled set accurately represents the underlying class distribution and contains classes that are not present in the labeled set. By doing so, we are able to transfer knowledge from few-shot to zero-shot classes without auxiliary information.

We assess the performance of our method on standard image classification datasets, reconstructing training sets where labels are obtained as described above. While varying the number of unseen classes, we also vary the conditions under which seen classes are sampled, including both balanced and imbalanced scenarios. Current SSL methods, after being adapted to identify unseen classes (see details in Section 3.4), are unable to successfully classify them. Our methodology, when incorporated with the same SSL methods, greatly improves their performance on unseen classes, while usually matching or even surpassing their performance on seen classes. Additionally, our method surpasses the overall performance of both open-set and open-world SSL methods, see Section 4.2. The improvement is especially high when the labeled set is small.

2 Previous Work

A number of related frameworks have been investigated in the machine learning community, motivated by problems that also inflict our scenario. **Table 2.1 summarizes the main characteristics of the most relevant frameworks**, highlighting the differences between them. Next, we discuss each one in more detail.

	Train labeled data		**Train unlabeled data**		**Test data**		**Expected output**	
Settings	**seen**	**unseen**	**seen**	**unseen**	**seen**	**unseen**	**seen**	**unseen**
†zero-shot learning	Y	N	-	-	Y/N	Y	classify	classify
*Semi-Supervised Learning (SSL)	Y	-	Y	Y/N	Y	-	classify	-
*Open set SSL (OSSL)	Y	N	Y	Y	Y	Y	classify	reject
Open world SSL (OWSSL)	Y	N	Y	Y	Y	Y	classify	classify
Our settings	few-shot	N	Y	Y	Y	Y	classify	classify

Table 2.1: Catalogue of different relevant frameworks as discussed in the text. The training set is divided into seen and unseen classes, a distinction not made in all frameworks. For each framework, the symbols 'Y' and 'N' denote that such data as specified in the column is permitted in the framework; 'Y' indicates that such data exists, and 'N' that it doesn't. '*' indicates frameworks in which unseen classes may be replaced with OOD data (see text). † indicates that additional semantic side-information such as attributes is needed. **Expected output:** Open-set SSL methods are not designed to classify new examples from unseen classes, only to reject them.

2.1 Closed and Open Semi-Supervised Learning

Closed and Open Semi-Supervised Learning , denoted **SSL** and **OSSL** respectively. SSL has made remarkable progress in recent years, see the comprehensive survey by Chen et al.

[4]. It involves leveraging sparsely labeled data and a considerable amount of additional unlabeled data, often sampled from the same underlying data distribution as the labeled data. In more general settings, Out Of Distribution (OOD) data may also be present in the unlabeled set.

In the original SSL settings [3], both the labeled and unlabeled data share a label set within the same domain. It is not, however, suitable for our scenario because it does not allow for unseen classes. Going a step further, open-set SSL (OSSL) permits the presence of unseen classes or OOD data in the unlabeled set. However, this framework is only designed to identify and reject data points that do not belong to the classes in the labeled set and is not tailored to classify unseen classes.

Some of the most effective SSL methods at present time combine ideas from self-supervised learning (e.g., consistency regularization), data augmentation, and the assignment of pseudo-labels to unlabeled data points, in order to utilize unlabeled data [e.g., 5, 18, 22, 28]. The field of Open-set SSL (OSSL) is also rapidly developing [8, 10, 14, 19, 26]. For comparison, we use the competitive OSSL method proposed by Saito et al. [17], which accomplishes the task by uniting the SSL and novelty detection frameworks.

Fixmatch [18] is a SSL algorithm often serving as a foundational framework for the development of various other algorithms in the field. The algorithm initially constructs a classifier using the available labeled data. It subsequently introduces an additional loss component based on data augmentation, specifically implementing a Consistency Regularization technique under input variation. This loss is computed using pseudo labels generated from the unlabeled data. The assignment of these pseudo labels is determined by applying a fixed confidence level threshold throughout the training process

Flexmatch [29] is a commonly utilized SSL algorithm that not only forms the core framework for our method but also serves as a reference baseline for comparison. In contrast to its predecessor, FixMatch, it's approach represents an advancement with dynamic threshold adjustments for assigning pseudo labels. These adjustments are executed at each step for each class, guided by the model's performance relative to other classes.

Freematch [22] is a SSL algorithm that serves as a reference baseline for our comparative analysis. In a specific scenario, Freematch forms the foundational framework for our method. It bears a resemblance to Flexmatch, with a distinguishing feature — Freematch calculates the dynamic threshold through an exponential moving average method that encompasses both a dynamic global threshold and a local, per-class dynamic threshold, the latter of which is intricately linked to the global threshold. Furthermore, a noteworthy augmentation to Freematch is the incorporation of a self-adaptive fairness loss component. This addition aids in aligning the algorithm's predictions with distributions akin to those observed in prior predictions

OpenMatch [17] is an OSSL algorithm, that serves as a reference baseline for our comparative analysis. This algorithm is designed to distinguish between seen and unseen classes using a novelty detection approach, facilitated by one-versus-all classifiers, with each classifier dedicated to distinguishing a specific seen class. It leverages the maximum confidence score for each sample to determine whether it belongs to one of the known seen classes or falls into the category of unseen classes. Additionally, Openmatch periodically generates pseudo labels for Fixmatch, a mechanism applied in the classification of samples determined by the algorithm to be seen classes. Another noteworthy aspect of Openmatch is its application of soft consistency regularization specifically for the one-versus-all classifiers on unlabeled data, integrating data augmentations similar to those employed in the Fixmatch framework.

2.2 Open-World SSL

Open-World SSL, denoted **OWSSL**. This paradigm is related to *Novel Class Discovery*, which aims to discover unseen classes within unlabeled data [9, 23, 24]. Joining SSL with this paradigm, OWSSL [2, 7, 16, 30] aims to discover unseen classes in an unlabeled set and to classify seen classes. Initially, OWSSL methods were transductive, in the sense that the test set is processed as a single batch, and classes are discovered by partitioning this set. Typically this is accomplished by learning pairwise similarities between the points in the unlabeled set.

In *OpenLDN*, Rizve et al. [16] extended the scope of the open-world methodology to our scenario. Specifically, they proposed to use the outcome, both seen classes and the partitioned unlabeled set, as input to an SSL method, thus obtaining a classifier for both seen and unseen classes. Our method differs from OpenLDA in that it solves the problem in an end-to-end fashion, and is thus more suitable for the low-budget regime with very few labeled examples. This methodology is not designed to receive a single test point and evaluate its class. Additionally, when given two sets of test data possibly at two different time points, its outcome may be inconsistent in the sense that different data partitions are not always easy to consolidate into a single coherent partition. When clustering is based on pairwise similarity, the resulting partition may be quite different when two groups of points are combined.

In contrast, our method provides a stable partition of the data, and a classifier that can identify the class (or cluster) of individual datapoints in a consistent manner and irrespective of other test points. In particular, it produces coherent results when given two separate test batches. The only uncertainty remains in the naming of clusters: if our unseen classes are dogs and cats, in the end our classifies identifies dogs as cluster 1 and cats as cluster 2 consistently, but without labels it cannot decide which cluster to call dogs and which one cats. To highlight this distinction, in Table 2.1 the first task (the partitioning of test data) is called 'class discovery', and the second 'classification'.

OpendLDN [16] is an OWSSL algorithm that serves as a reference baseline for our comparative analysis. OpenLDN utilizes a pairwise similarity loss to discover novel classes. Using a bi-level optimization rule this pairwise similarity loss exploits the information available in the labeled training data to implicitly cluster novel class samples, while simultaneously recognizing samples from known classes. After discovering novel classes, OpenLDN transforms the open-world SSL problem into a standard SSL problem by creating pseudo labels for the unseen classes and refer them as part of the labeled training data.

Zero-Shot Learning, denoted **ZSL**. Work on ZSL often involves the computation of a separate label embedding space and assumes access to semantic data that may be difficult to obtain. Some recent work aims to expand the scope of this methodology to domains where the seen classes are only sparsely sampled. For example, Li et al. [12], Xu et al. [25] leverage SSL in order to aid the computation of label embedding, in a two-step manner.

2.3 Summary of contribution

We present and justify a novel SSL variation, characterized by classes that are present in both the unlabeled set and the test set, yet absent from the labeled set. Addressing this variant, we offer a solution that:

i. Outperforms all current baselines by a large margin.

ii. Can be assimilated into any state-of-the-art SSL method.

iii. In few-shot settings, the combined solution converges significantly faster than its underlying SSL method.

3 Methods

Traditionally, different metrics are used to evaluate performance when dealing with labeled or unlabeled data. With labeled data, metrics that measure classification accuracy are appropriate. With unlabeled data, the outcome is essentially a partition of the data – no specific labels are assigned to specific clusters. Here, the appropriate metrics measure the partition accuracy, as used to evaluate clustering algorithms. These scores are detailed in Section 3.2.

In Section 3.3 we describe our method, designed to learn a classifier for the Semi-Supervised Learning scenario with few-shot and zero-shot classes. In Section 3.4 we discuss how baseline methods can be used for comparison. Since most of the relevant frameworks (see discussion above) are not designed to classify unseen classes, we describe an adaptation step, which allows us to obtain unseen class classification from SSL and open-set SSL methods.

3.1 Notations and definitions

Let $\mathcal{X}$ denote the set of all possible examples, and C the set of possible classes. We consider a learner denoted as $f : \mathcal{X} \to C$, which has access to a labeled set of examples $\mathcal{L} \subseteq \mathcal{X} \times C$ and an unlabeled set of examples $\mathcal{U} \subseteq \mathcal{X}$. In our scenario, the labeled examples in $\mathcal{L}$ do not contain all possible classes of C. Let $C_{seen} \subseteq C$ denote the set of classes that appears in $\mathcal{L}$, and $C_{unseen} \subseteq C$ the set of classes that do not appear in $\mathcal{L}$. Note that $C_{seen} \cap C_{unseen} = \emptyset$, and $C_{seen} \cup C_{unseen} = C$. For evaluation purposes, we assume the existence of a test set $\mathcal{T} \subseteq \mathcal{X} \times C$, which is disjoint from both $\mathcal{U}$ and $\mathcal{L}$. $\mathcal{T}$ contains examples from all classes. We denote the examples from seen classes in the test set as $\mathcal{T}_{seen}$ and the examples from unseen classes as $\mathcal{T}_{unseen}$.

3.2 Evaluation scores

Open-world SSL scores. As customary in novel class discovery and OWSSL, the following scores are used:

1. *seen classes accuracy*

$$acc_{seen} = \frac{1}{|\mathcal{T}_{seen}|} \sum_{x_i \in \mathcal{T}_{seen}} \mathbb{1}_{[f(x_i)=y_i]}$$

2. *unseen classes accuracy*

$$acc_{unseen} = \frac{1}{|\mathcal{T}_{unseen}|} \sum_{x_i \in \mathcal{T}_{unseen}} \mathbb{1}_{[f(x_i)=y_i]}$$

   This score is based on the best permutation of the unseen classes – similarly to what is done in clustering accuracy.

3. *combined score*

$$\frac{|C_{seen}|}{C} \cdot acc_{seen} + \frac{|C_{unseen}|}{C} \cdot acc_{unseen}$$

Open-set SSL scores. In OSSL, where the goal is only to reject unseen classes and not to classify them, different scores are used. For a fair comparison, we report these scores below in cases suitable for the OSSL framework, with many missing classes and sufficient labels for each seen class.

1. *closed accuracy*

$$\frac{1}{|\mathcal{T}_{seen}|} \sum_{x_i \in \mathcal{T}_{seen}} \mathbb{1}_{[f|_{seen}(x_i)=y_i]}$$

   Here, $f|_{seen}$ is the prediction of f when forced to choose a class out of C_{seen}. Thus an OSSL method is not allowed to reject. Other methods are not allowed to classify points for labels in C_{unseen}.

2. *unknown accuracy*

$$\frac{1}{|\mathcal{T}_{unseen}|} \sum_{x_i \in \mathcal{T}_{unseen}} \mathbb{1}_{[f(x_i)\in C_{unseen}]}$$

   This score captures the rejection accuracy.

3. *AUROC* – Area under ROC curve of 'reject' classifier.

Note the difference between *seen classes accuracy* and *closed accuracy*: the first penalizes for *seen* points recognized as *unseen*, while the second doesn't, and therefore *seen classes accuracy* is always lower than *closed accuracy*.

3.3 Our Method

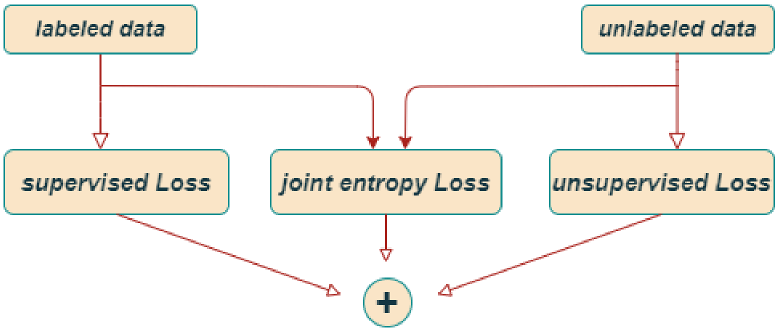


Figure 3.1: The combined SSL loss used in our approach.

We denote as B the batch size of the learner. Let $\overline{p}_i$ denote the empirical probability of class $i \in C$ across all the classes, in a specific batch:

$$\overline{p}_i = \frac{1}{B} \sum_{b=1}^{B} f_i(x_b)$$

Where f_i is the confidence of f on class $i \in C$. We define the following entropy loss:

$$\ell_e = - \sum_{i \in C_{seen} \cup C_{unseen}} \overline{p}_i \log(\overline{p}_i)$$

This choice assumes that all classes are roughly equally distributed in the data. If the prior distribution of class frequencies is known to be the vector $\{g_i\}_i$, we will use instead the KL divergence between the probability vectors:

$$\ell_e = \sum_{i \in C_{seen} \cup C_{unseen}} \overline{p}_i \left[\log(g_i) - \log(\overline{p}_i)\right]$$

Most SSL methods optimize a loss function which is essentially the sum of an unsupervised term ℓ_u and supervised term ℓ_s. In our approach, we adopt the full pipeline of the method, while optimizing the following objective function:

$$\ell = \ell_s + \ell_u + \lambda \ell_e$$

This design is illustrated in Fig. 3.1.

Model Collapse

During the training of our algorithm, we sometimes encounter a phenomenon in which the confidence of the model's predictions falls drastically, especially the confidence of points from unseen classes. This phenomenon results in the model being essentially unable to generate pseudo labels, and therefore the scores show a sharp sudden decline. This may happen relatively early in the training as there are fewer labels per class.

Our solution is to look at the entropy loss during training and to identify any sharp decline (using a threshold on the gradient graph). Once identified, a stopping point is fixed a few epochs before the sudden decline. An Interesting outcome is that our algorithm converges much faster than its original SSL variant. For example, with 4 labels per class and 40 unseen classes in CIFAR-100, our method converges after 523.0 ± 24.61 epochs, while the SSL methods of *FlexMatch* and *FreeMatch* require 1024 epochs, *OpenMatch* - 512, and *OpenLDN* - 1024.

3.4 Baseline methods for comparison

Semi-Supervised Learning in the Few-Shot Zero-Shot Scenario is a somewhat new setting, and there are no ready-to-use baseline methods that are designed to solve it effectively. Consequently, we extend methods designed to deal with closely related problems, as discussed in the introduction (see Table 2.1), and report their results. It should be kept in mind that since the methods are designed for other scenarios, their inferior results here should not be taken as evidence that they are not suited for their original task, where they are likely to achieve state-of-the-art results.

More specifically, we adapt methods designed to deal with the following scenarios: (i) semi-supervised learning (SSL), (ii) open-set SSL (OSSL), (iii) open-world SSL (OWSSL). We start by describing how to extend SSL and OSSL methods to the OWSSL scenario, similarly to how it's done by Cao et al. [2]. We then conclude by discussing the adaptation of the OWSSL scenario.

Traditional SSL methods are designed to classify seen classes, but they do not expect to encounter unseen classes in the test set. In order to extend them to the OSSL scenario, we adopt a common way to add a reject option to classifiers [e.g., 20], by applying a threshold test to the softmax confidence score of the model. To compensate for the simplicity of the approach, we choose for each method its optimal threshold based on ground truth information. In the end, we have a set of points classified into seen classes, and a set of rejected points, similar to the OSSL scenario to be discussed next.

Open-set SSL methods are designed to classify seen classes and reject unseen classes, but can't classify unseen classes. In order to extend these methods to our scenario, an additional step is employed to partition the rejected points to $|C_{unseen}|$ parts. This is accomplished with K-means clustering, performed over the model's feature space.

As customary for unlabeled data, and for the purpose of evaluation only, the set of cluster labels is matched to the set of unseen labels C_{unseen} using the best permutation. Note that the *unseen classes accuracy* score above employs a similar procedure. As a result, the set of test points is now classified to all the labels in C.

Open-world SSL methods are designed to process the labeled and unlabeled training sets and output a partition of the unlabeled set. This partition is matched to C_{unseen} as described above. Finally, in order to generate a classifier for future unseen points, we follow the procedure suggested by Rizve et al. [16]: use the labeled training set as is and the unlabeled set with its inferred labels to train another classifier, whose domain of output labels is C.

4 Empirical Results

We now describe the empirical settings used to evaluate our method in comparison to established baselines.

4.1 Methodology and technical details

Datasets. We use two benchmark image datasets in the experiments below: CIFAR-100 [11] and STL-10 [6]. When using STL-10, we omitted the unlabeled split due to its inclusion of out-of-distribution examples.

Baselines. We compare our method to the following baselines: (i) SSL - *FreeMatch* [22] which performs well in the few-shot regime, and *FlexMatch* [28] which is the backbone of our own method. (ii) Open-set SSL - *OpenMatch* [17]. (iii) Open-world SSL: *OpenLDN* [16], which includes its own adaptation to our scenario using *MixMatch* [1]. For *FlexMatch* and *FreeMatch* we employ the SSL evaluation framework established by [21], ensuring a fair comparison. For *OpenMatch* and *OpenLDN* algorithms, we used the source code provided with the original papers. The methods were adapted to our scenario as detailed in Section 3.4.

Architectures and Hyper-parameters. When training *FlexMatch*, *FreeMatch*, and *OpenMatch*, we considered a Wide-ResNet-28 (WRN) [27], trained with stochastic gradient descent optimizer, 64 batch size, 0.03 learning rate, 0.9 momentum and 5e-4 weight decay. In Fig 4.1 we used 8 width factor, as was done in the original papers. Due to its heavy computational cost, in all other experiments, we reduced the width factor to 2. We note that the qualitative results remained the same.

When training *OpenLDN* , we used the official code, composed of a basic model and a *MixMatch* model. Both models use ResNet-18, trained with Adam optimzier. The base model uses a 200 batch size and a 5e-4 learning rate. *MixMatch* uses 64 batch size and 0.002 learning rate.

The λ for ℓ_e was set to 1.5 in all experiments. We used NVIDIA GeForce RTX 2080 GPUs for all experiments.

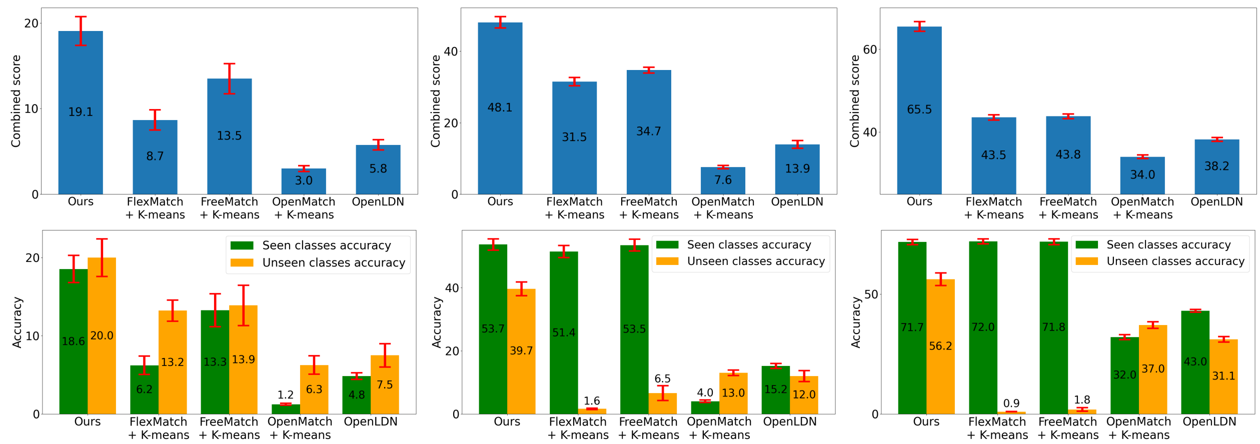
Split of classes to seen and unseen. When training on CIFAR-100, the split of the 100 classes into seen and unseen classes was done randomly, repeating the exact same partition for all models. It's important to note that in the empirical evaluation of *OpenMatch*, outlined in [17], the choice of unseen classes was determined by their super-class membership, introducing a potential bias. This may explain the differences in accuracy between the results we report below for *OpenMatch*, and those reported in the paper. To validate this point, we used their split method in a subset of our experiments, those involving the many-shot scenarios with 100 or 250 examples per class.

4.2 Main Results

Balanced labeled set, few-shot for seen classes. We begin by comparing the performance of our method, combined with *FlexMatch*, with two SSL methods – *FlexMatch* and *FreeMatch*, one OSSL method – *OpenMatch*, and one OWSSL method – *OpenLDN*. *FlexMatch*, *FreeMatch*, and *OpenMatch* are adapted with k-means to our scenario, as described in Section 3.4. All methods are trained on CIFAR-100, with 40 unseen classes (similar qualitative results are obtained when the number of classes is varied, see ablation study). The number of labeled points in each seen class varies, from one-shot with 1 example, few-shot with 4 examples and mid-shot with 25 labels, which can be found in Figs. 4.1a,4.1b,4.1c, respectively.

Inspecting the *combined accuracy* score in Fig. 4.1, we observe that our method outperforms the other baselines by large margins. In each case, we also plot the *seen* and *unseen classes accuracy* for each method. Evidently, the improvement over other SSL algorithms is caused primarily by their poor performance on unseen classes, which in turn is caused by their overconfidence in their erroneous classification of seen classes. This is a well-known problem [13] when attempting to add a reject option to deep models, which can be further appreciated from the comparison in the ablation study, Fig. 4.5.

It's worth noting that the subpar performance of *OpenMatch* and *OpenLDN* can be primarily attributed to their intended focus on scenarios with considerably higher availability of la-



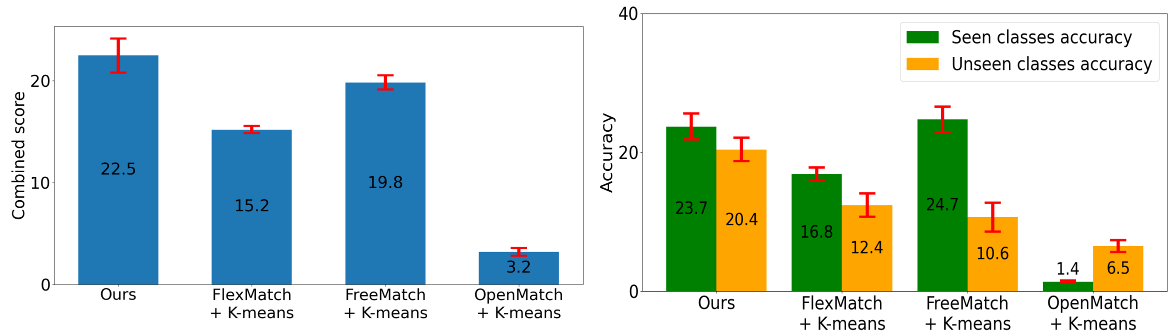
(a) 1 label per class, 35 unseen classes (b) 4 labels per class, 40 unseen classes (c) 25 labels per class, 40 unseen classes

Figure 4.1: Comparison of our method to the baselines discussed in Section 4.1. All methods are trained on CIFAR-100, where (a),(b), and (c) differ in the size of the labeled set and the number of unseen classes. In the top row, we show the *combined score* of each method. In the bottom row, we show the *seen* and *unseen accuracy* of each method. We clearly see that our method outperforms all baselines by significant margins. When compared to SSL methods - *FreeMatch* and *FlexMatch*, we see that while the performance on the seen classes is comparable, our method improves the performance on the unseen classes quite drastically. Both *OpenMatch* and *OpenLDN* are not suitable for the few-shot regime, and their performance is therefore low.

beled data. Their under-performance stems from being tailored to suit different tasks, which are evaluated using different metrics as discussed in Section 3.2.

Non-uniform labeled set, approximately one-shot We now get back to the original scenario, where the labels are obtained as follows: To begin with, the learner is given a large set of unlabeled data cheaply collected. Additionally, there is a fixed budget that allows for the labeling of only n examples. The learner randomly chooses n examples from the unlabeled set to be labeled, thus obtaining the labeled training set. The labeled set is quite imbalanced, with a range of examples for each seen class, from 1 up to 3 or even 4. The remaining examples compose the unlabeled training set.

Results of this setup are shown in Fig. 4.2, for CIFAR-100 and $n = 100$ labeled examples. Once again, when inspecting the *combined score*, our method outperforms all other baselines. Given the technical difficulties that this unusual setup poses, we could not obtain reliable

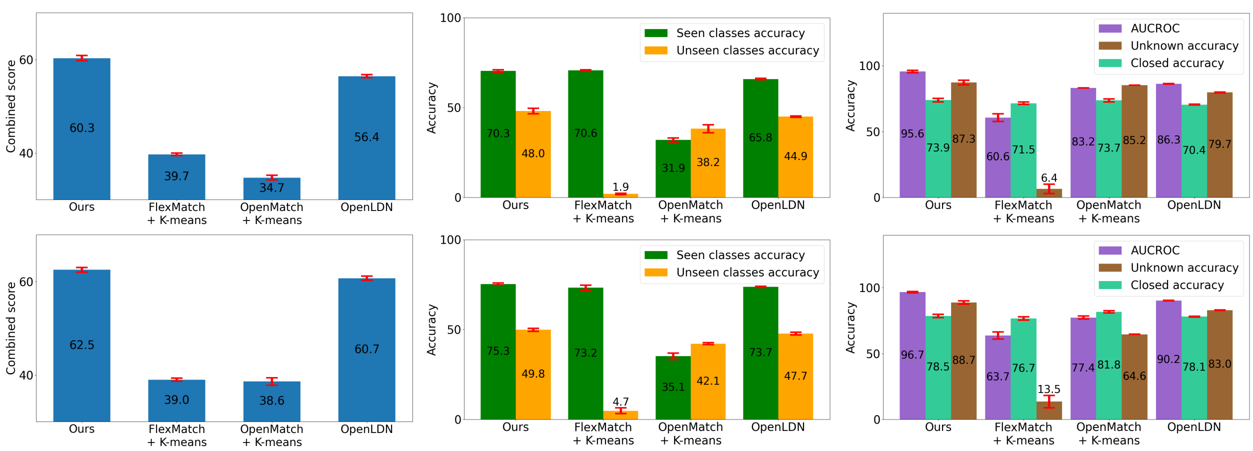


(a) *Combined score* (b) *Seen* and *unseen accuracy*

Figure 4.2: CIFAR-100, 100 labeled examples sampled at random (see text). On average, there are 36.61 ± 1.38 unseen classes, and 1.58 ± 0.03 labeled examples per seen class.

results with *OpenLDN*. Nevertheless, we expect *OpenLDN* to perform poorly in these conditions, as seen with 1 labeled example per seen class in Fig. 4.1a.

Balanced labeled set, many-shot for seen classes. In our final setting, we increase the number of labeled examples per class, thus obtaining scenarios that are more favorable to the alternative methods, especially *OpenMatch* and *OpenLDN*. To shapren the comparison with these methods, we add the 3 valuation scores discussed in Section 3.2, which are sometimes used to evaluate open-set SSL methods. Results[1] of this setup with 5 repetitions are shown in Fig. 4.3. Our method significantly outperforms all baselines, though the gap with *OpenLDN* is smaller.

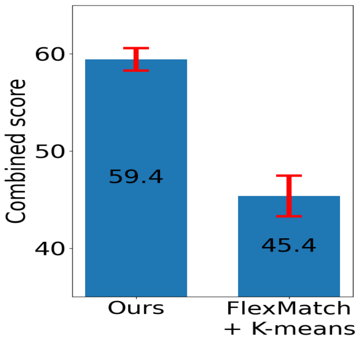


(a) *Combined score* (b) *Seen* and *unseen accuracy* (c) OSSL scores

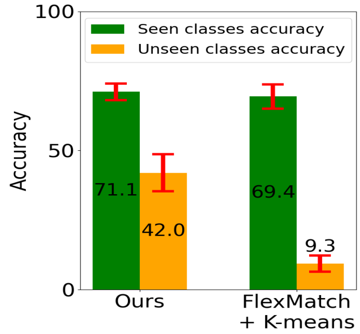
Figure 4.3: Top: CIFAR-100, 100 labels per class, 45 unseen classes. Bottom: CIFAR-100, 250 labels per class, 50 unseen classes. Split is based on super-classes, as explained in Section 4.1. The various scores, including the additional OSSL scores, are described in Section 3.2.

[1]Since *FlexMatch* and *FreeMatch* perform similarly in the many-shot case, we only show results using *FlexMatch*.

4.3 Ablation Study



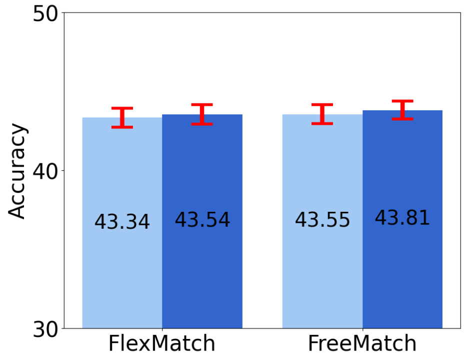
(a) *Combined score*



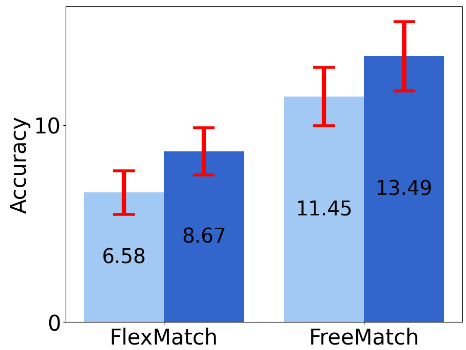
(b) *Seen* and *unseen accuracy*

Figure 4.4: STL-10, 4 labels per class, 4 unseen classes.

Balanced labeled set, few-shot for seen classes, STL-10. To verify that the results reported above are not unique to a single dataset, we repeated some of the experiments with an inherently different image dataset - STL-10, showing similar qualitative results. We evaluate a case with very few labeled examples - 4 labeled examples for each of 6 seen classes, and 4 unseen classes. Since *OpenMatch* and *OpenLDN* are not natively adapted to STL-10, we limited the comparison of our method, which is based on *FlexMatch*, to that of *FlexMatch* without out method. The results are shown in Fig. 4.4.



(a) 25 labels per class, 40 unseen classes



(b) 1 label per class, 35 unseen classes

Figure 4.5: *Combined score* of *FlexMatch* and *FreeMatch* with and without the K-means adaptation (see Section 3.4), on CIFAR-100. *With K-means* - dark blue, *Without K-means* - light blue.

SSL adaptation. To adapt the SSL methods to our scenario, we introduced in Section 3.4 a two-step adaptation process: (i) Points with low classification confidence are rejected from the classifier trained on seen classes. (ii) K-means clustering is used to partition the rejected points within the feature space. This partitioning is then matched to unseen classes. Without this adaptation, SSL methods would not identify anything as unseen, and would therefore erroneously match all points from unseen classes to some seen class. In each case, the rejection threshold was optimized to give each method its best possible *combined score*.

Here we evaluate the added value of this adaptation, as shown in Fig. 4.5. Evidently, the impact is minor (especially in the mid-shot regime). The reason is that the optimal threshold tends to be relatively low, resulting in a relatively modest count of predicted unseen instances. This count often pales in comparison to the actual prevalence of unseen classes within the test data. The underlying reason for this failure is the known property of deep models to be over-confident in their predictions. It is therefore hard to distinguish between seen and unseen classes based on the confidence threshold.

Convergence time As described in detail in Section 3.3, sometimes the SSL training, performed by our method, is stopped relatively early. This happens when the entropy loss suddenly declines sharply, and is usually correlated with small numbers of labeled examples per seen class. As a result, it so happens that our model needs much fewer epochs to converge, as compared to the original model that serves as its backbone, *FlexMatch*. This result is shown in Table 4.1.

Labels per class	Unseen classes	Convergence epoch
1	35	224.6 ± 44.19
4	40	523.0 ± 24.61
25	40	726.6 ± 8.85

Table 4.1: The mean number of epochs to convergence, in a scenario where *FlexMatch* needs 1024 epochs to converge.

5 Conclusion

We investigated Semi-Supervised Learning (SSL) in a small sample framework with few-shot and zero-shot classes, thereby unveiling an unexplored real-life challenge. To address this challenge, we proposed to integrate an entropy loss into established state-of-the-art SSL frameworks, exemplified by *FlexMatch*. This strategic enhancement serves as a potent remedy for the absence of labeled instances corresponding to unseen classes. This method will enable us to convert any state-of-the-art SSL algorithm to readily face this challenge.

Our extensive empirical evaluation substantiates the efficacy of our approach in resolving the aforementioned challenge when juxtaposed against alternative baselines, including SSL and open-set SSL (adapted to suit our scenario as discussed in Section 3.4), as well as open-world SSL. Notably, our algorithm emerges as a standout performer, particularly within the realm of limited labeled data scenarios.

In future work, we will investigate the dynamic incorporation of the entropy loss throughout the training process. This adaptive integration holds promise for enhancing the robustness and versatility of our approach.

Bibliography

[1] Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. A. (2019). Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32.

[2] Cao, K., Brbic, M., and Leskovec, J. (2022). Open-world semi-supervised learning. In *International Conference on Learning Representations (ICLR)*.

[3] Chapelle, O., Scholkopf, B., and Zien, A. (2009). Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542.

[4] Chen, Y., Mancini, M., Zhu, X., and Akata, Z. (2022). Semi-supervised and unsupervised deep visual learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*.

[5] Chen, Y., Zhu, X., Li, W., and Gong, S. (2020). Semi-supervised learning under class distribution mismatch. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 3569–3576. AAAI Press.

[6] Coates, A., Ng, A., and Lee, H. (2011). An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings.

[7] Guo, L.-Z., Zhang, Y.-G., Wu, Z.-F., Shao, J.-J., and Li, Y.-F. (2022). Robust semi-supervised learning when not all classes have labels. *Advances in Neural Information Processing Systems*, 35:3305–3317.

[8] Guo, L.-Z., Zhang, Z.-Y., Jiang, Y., Li, Y.-F., and Zhou, Z.-H. (2020). Safe deep semi-

supervised learning for unseen-class unlabeled data. In *International Conference on Machine Learning*, pages 3897–3906. PMLR.

[9] Han, K., Rebuffi, S., Ehrhardt, S., Vedaldi, A., and Zisserman, A. (2020). Automatically discovering and learning new visual categories with ranking statistics. In *Proceedings of the 8th Intennational Conference on Learning Representations, ICLR 2020*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

[10] Huang, J., Fang, C., Chen, W., Chai, Z., Wei, X., Wei, P., Lin, L., and Li, G. (2021). Trash to treasure: Harvesting ood data with cross-modal matching for open-set semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8310–8319.

[11] Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

[12] Li, X., Guo, Y., and Schuurmans, D. (2015). Semi-supervised zero-shot classification with label representation learning. In *Proceedings of the IEEE international conference on computer vision*, pages 4211–4219.

[13] Nguyen, A., Yosinski, J., and Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436.

[14] Park, J., Yun, S., Jeong, J., and Shin, J. (2022). Opencos: Contrastive semi-supervised learning for handling open-set unlabeled data. In *European Conference on Computer Vision*, pages 134–149. Springer.

[15] Pourpanah, F., Abdar, M., Luo, Y., Zhou, X., Wang, R., Lim, C. P., Wang, X.-Z., and Wu, Q. J. (2022). A review of generalized zero-shot learning methods. *IEEE transactions on pattern analysis and machine intelligence*.

[16] Rizve, M. N., Kardan, N., Khan, S., Shahbaz Khan, F., and Shah, M. (2022). Openldn: Learning to discover novel classes for open-world semi-supervised learning. In *European Conference on Computer Vision*, pages 382–401. Springer.

[17] Saito, K., Kim, D., and Saenko, K. (2021). Openmatch: Open-set consistency regularization for semi-supervised learning with outliers. *CoRR*, abs/2105.14148.

[18] Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., Cubuk, E. D., Kurakin, A., and Li, C.-L. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608.

[19] Sun, Y. and Li, Y. (2023). Opencon: Open-world contrastive learning. *Trans. Mach. Learn. Res.*, 2023.

[20] Tax, D. M. and Duin, R. P. (2008). Growing a multi-class classifier with a reject option. *Pattern Recognition Letters*, 29(10):1565–1570.

[21] Wang, Y., Chen, H., Fan, Y., Sun, W., Tao, R., Hou, W., Wang, R., Yang, L., Zhou, Z., Guo, L.-Z., Qi, H., Wu, Z., Li, Y.-F., Nakamura, S., Ye, W., Savvides, M., Raj, B., Shinozaki, T., Schiele, B., Wang, J., Xie, X., and Zhang, Y. (2022). Usb: A unified semi-supervised learning benchmark for classification. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[22] Wang, Y., Chen, H., Heng, Q., Hou, W., Fan, Y., Wu, Z., Wang, J., Savvides, M., Shinozaki, T., Raj, B., Schiele, B., and Xie, X. (2023). Freematch: Self-adaptive thresholding for semi-supervised learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

[23] Wang, Z., Salehi, B., Gritsenko, A., Chowdhury, K., Ioannidis, S., and Dy, J. (2020). Open-world class discovery with kernel networks. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 631–640. IEEE.

[24] Xie, J., Girshick, R., and Farhadi, A. (2016). Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR.

[25] Xu, B., Zeng, Z., Lian, C., and Ding, Z. (2021). Semi-supervised low-rank semantics grouping for zero-shot learning. *IEEE Transactions on Image Processing*, 30:2207–2219.

[26] Yu, Q., Ikami, D., Irie, G., and Aizawa, K. (2020). Multi-task curriculum framework for open-set semi-supervised learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*, pages 438–454. Springer.

[27] Zagoruyko, S. and Komodakis, N. (2016). Wide residual networks. *arXiv preprint arXiv:1605.07146*.

[28] Zhang, B., Wang, Y., Hou, W., Wu, H., Wang, J., Okumura, M., and Shinozaki, T. (2021a). Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 18408–18419.

[29] Zhang, B., Wang, Y., Hou, W., Wu, H., Wang, J., Okumura, M., and Shinozaki, T. (2021b). Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling.

[30] Zhong, Z., Zhu, L., Luo, Z., Li, S., Yang, Y., and Sebe, N. (2021). Openmix: Reviving known knowledge for discovering novel visual categories in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9462–9470.