PLMCL: Partial-Label Momentum Curriculum Learning for Multi-Label Image Classification

Rabab Abdelfattah[1], Xin Zhang[1], Zhenyao Wu[2], Xinyi Wu[2]
Xiaofeng Wang[1], and Song Wang[2]

[1] Department of Electrical Engineering, University of South Carolina, USA
[2] Department of Computer Science, University of South Carolina, USA
{rabab,xz8,zhenyao,xinyiw}@email.sc.edu, {wangxi, songwang}@cec.sc.edu

Abstract. Multi-label image classification aims to predict all possible labels in an image. It is usually formulated as a partial-label learning problem, given the fact that it could be expensive in practice to annotate all labels in every training image. Existing works on partial-label learning focus on the case where each training image is annotated with only a subset of its labels. A special case is to annotate only one positive label in each training image. To further relieve the annotation burden and enhance the performance of the classifier, this paper proposes a new partial-label setting in which only a subset of the training images are labeled, each with only one positive label, while the rest of the training images remain unlabeled. To handle this new setting, we propose an end-to-end deep network, PLMCL (Partial-Label Momentum Curriculum Learning), that can learn to produce confident pseudo labels for both partially-labeled and unlabeled training images. The novel momentum-based law updates soft pseudo labels on each training image with the consideration of the updating velocity of pseudo labels, which help avoid trapping to low-confidence local minimum, especially at the early stage of training in lack of both observed labels and confidence on pseudo labels. In addition, we present a confidence-aware scheduler to adaptively perform easy-to-hard learning for different labels. Extensive experiments demonstrate that our proposed PLMCL outperforms many state-of-the-art multi-label classification methods under various partial-label settings on three different datasets.

1 Introduction

With the advances in deep learning, significant progress has been made on single-label image classification problems [11] where each image only has one label. However, in many real applications, one image may contain multiple objects and/or exhibit multiple attributes that cannot be well described by a single label. For instance, a scene may contain multiple objects and a CT scan may indicate multiple conditions. This leads to an important computer-vision task of multi-label image classification that aims to identify all the labels on an image. A great challenge in multi-label image classification comes from the need of a large number of labeled training images. In particular, many supervised-learning

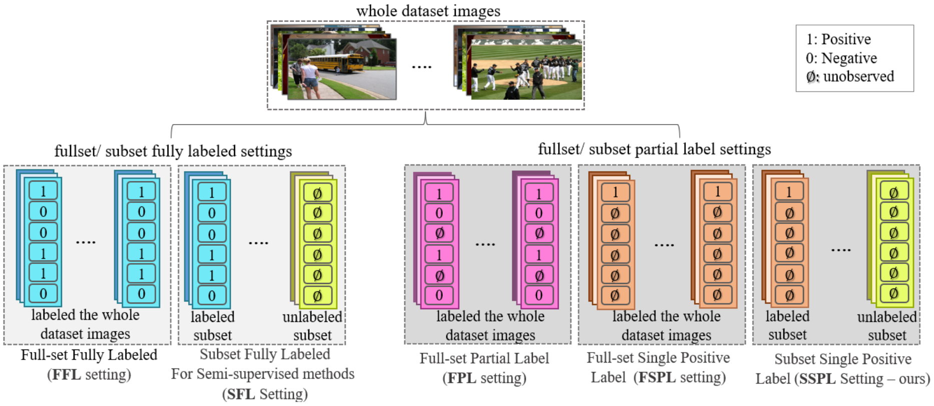
whole dataset images
1: Positive
0: Negative
∅: unobserved
fullset/ subset fully labeled settings
fullset/ subset partial label settings
labeled the whole dataset images
labeled subset
unlabeled subset
labeled the whole dataset images
labeled the whole dataset images
labeled subset
unlabeled subset
Full-set Fully Labeled (FFL setting)
Subset Fully Labeled For Semi-supervised methods (SFL Setting)
Full-set Partial Label (FPL setting)
Full-set Single Positive Label (FSPL setting)
Subset Single Positive Label (SSPL Setting – ours)

Fig. 1: An illustration of various label settings. **Full-set fully labeled (FFL)**: All positive and negative labels are annotated on each training image. **Full-set partial labels (FPL)** [13]: At least one positive or negative label is annotated on each training image. **Full-set single positive labels (FSPL)** [10]: Exactly one positive label is observed on each training image. **Subset fully labeled (SFL)**: All positive and negative labels are annotated on a subset of training images and the rest of the images are unlabeled. **Subset single positive labels (SSPL-ours)**: Exactly one positive label is annotated on each image from a subset of the training dataset while the rest of the images are unlabeled.

algorithms require all labels on every training image to be accurately annotated, which can be very difficult and laborious [12].

To relieve the annotation burden of fully labeling, recent works on multi-label classification consider training the network with partial labels [10,13,17,22,42,29]. One typical partial-label setting is **full-set partial labels (FPL)** [13], which annotates only a subset of labels on each training image. Following this setting, a special case is **full-set single positive labels (FSPL)** [10], which annotates only one positive label on each image (Fig. 1). Although these partial-label settings can, to some extent, mitigate the annotation burden, we still need to go through all training images for annotation.

To go one step further, we consider a new partial-label setting of **subset single positive labels (SSPL)**, where the training dataset consists of both labeled and unlabeled images, while, for the subset of labeled images, only one positive label is annotated on each image. Clearly, this new setting can further reduce the annotation cost for large-scale multi-label datasets and practically our setting can be utilized by only annotating a limited percentage of total images with one single positive label per image. For instance, COCO dataset has 2.5M labels which takes 20K worker hours to identify categories [24], while 60% SSPL only requires 49.2k labels and the annotation time is up to 394 hours. Here we did not even account for the fact that detecting absence can be more difficult than detecting presence. However, the inclusion of unlabeled images significantly increases the difficulty of the network design and training. In particular, those existing partial-label learning methods based on label correlation [17] and label matrix completion [5] are not applicable to our new setting since they cannot handle single positive labels and unlabeled data simultaneously. From another point of view, although our proposed setting and semi-supervised learning both use unlabeled subsets, the performance of semi-supervised models may degrade when training on the SSPL setting. This is because semi-supervised methods usually assume access to a subset of fully-labeled images (SFL setting) to initialize the training process, while such a fully-labeled subset is not available on the SSPL setting as shown in Fig. 1. Such degradation can be observed in the comparison experiments presented in Section 4.4.

This paper presents a new Partial-Label Momentum Curriculum Learning (PLMCL) method for end-to-end training of the multi-label classifier under the proposed new partial-label setting. In particular, a set of soft pseudo labels for each image are estimated and dynamically updated by momentum. The momentum combines the classifier predictions and the pseudo labels to identify the direction for updating the pseudo labels. Meanwhile, to update the multi-label classifier parameters, we introduce a scheduled loss function based on standard cross-entropy with which the learning gradually moves from observed (easy) labels to unobserved (hard) labels. Our proposed PLMCL can also be easily extended to handle other partial-label settings [10,13,17] for multi-label image classification.

Our contributions are summarized as follows:

- We introduce a new partial-label setting, SSPL, for multi-label image classification by allowing the inclusion of unlabeled images as training images. This setting can be as simple as the case where only a subset of images have labels and only one single positive label is annotated per image in this subset. This new setting can significantly reduce the annotation burden as well as leverage more unlabeled data for multi-label image classification.
- We propose a new PLMCL framework for multi-label image classification under different partial-label settings. The major novelty comes from the introduction of the momentum and the confidence levels into the partial-label setting. Instead of accelerating learning as discussed in [15], the momentum in this paper actually brings a stronger stability requirement into training, which focuses on the convergence of not only the pseudo labels, but also the updating velocity of pseudo labels that indicates high confidence on the pseudo labels. This is important, especially at the early stage of training when the observed labels are lack and the confidence levels on the pseudo labels are low. Otherwise, premature convergence could be achieved with low confidence levels due to lack of observed labels.
- Extensive experimental results show that our method outperforms the state-of-the-arts under three different partial-label settings on three widely-used datasets. With fewer observed labels in our proposed setting, our method can still get comparable classification results as those methods using full-set single positive labels (FSPL) setting, as shown in Fig. 1.

The rest of the paper is organized as follows. Section 2 briefly reviews the related works. Section 3 presents the details of the proposed method. The experiment results are demonstrated in Section 4. Finally, the conclusions are drawn in Section 5.

2 Related Work

This section briefly reviews the related work on partial-label learning under different settings (Fig. 1) as well as semi-supervised learning and curriculum learning methods.

2.1 Partial-label Learning under Different Settings

Full-set Partial Labels (FPL). Early work on multi-label image classification under the FPL setting assumes that the unobserved labels are negatives [33,27,4,7]. This assumption inevitably introduces false negative labels during the training and therefore will lead to a significant performance degradation in classification. Another direction is to estimate the unobserved labels based on the correlation between the labels [27,41], label matrix completion [39,5], low-rank learning [44,43], and probabilistic models [20,8]. Most of these models, however, have to solve an optimization problem during the training, which might be computationally expensive for deep learning given the high complexity of deep neural networks [13]. Some recent work uses end-to-end deep learning models to predict unobserved labels. For instance, the method in [17] takes advantage of image/label similarity graphs and builds the dependencies among the labels based on the label co-occurrence information. Nevertheless, it requires more than one label per image to build these relations and therefore is not applicable for single-label settings. Another work in [13] integrates graph neural network with curriculum learning. The proposed strategy predict the unobserved labels based on network models and add those predicted labels into the training dataset using a threshold-based strategy. Those selected labels are called "weakly labels", which may change over epochs. Therefore, there is a possibility that the predictions of some unobserved labels are never selected into the training, which may result in information loss. In contrast, our model learns the pseudo labels dynamically based on the momentum throughout the training process. Different from [13], every pseudo label will more or less contribute to the training in each epoch, depending on its confidence level. This can guarantee the continuity in pseudo label updates, which is lack from the threshold-based strategy.

Full-set Single Positive Label (FSPL). This setting is introduced in [10], where the authors present an end-to-end model that jointly trains the image classifier and the label estimator for online estimation of unobserved labels. The label estimator is implemented to build a simple look-up table which requires to store a matrix containing the whole dataset images, including all the previously estimated labels per image, in memory and update it in each epoch during the entire training process. Obviously, this method requires ample memory space to store the full label matrix for all training images, which makes it almost infeasible for large datasets or large numbers of labels. Moreover, this method relies on random initialization for the label estimator and assigns equal weights for the observed and unobserved terms in the suggested loss function, which may lead to degradation in the classifier's performance. In contrast, our method initializes the pseudo labels with unbiased probability values guaranteeing stable training using a self-guided momentum factor. Therefore, the self-guided factor works to (i) identify the amount of change should the labels get to move towards 1 or 0 starting from unbiased probability, and (ii) reduce the amount of change for the labels with a high confidence score. In addition, our momentum-based updating method does not require keeping the full labels in the memory since we only keep B vectors in memory per each batch, where B is the batch size. Finally, our method uses a scheduled loss function to give different weights for the unobserved term, which can gradually move the learning from the observed labels to the unobserved labels.

Subset Single Positive Labels (SSPL). This is our proposed setting which assumes that each labeled image is annotated with only a single positive label, while the unlabeled images are totally unlabeled. A practical scenario is iNaturalist dataset, where each image only has one positive label while the other classes are unlabeled [10]. Adding new unlabeled images to enrich this dataset directly leads to the SSPL setting. Actually, this applies to most datasets for multi-class classification. Adding new unlabeled images makes those datasets suitable for the multi-label problem under the SSPL setting without annotation costs. To the best of our knowledge, there are no approaches specifically designed for SSPL. However, some semi-supervised models can be modified to fit SSPL setting. We will go through these methods in the next subsection. Although these methods [22,26,10] can be easily applied to the SSPL setting, these methods may still not handle the unlabeled set of the SSPL properly. Since these models usually assign assumptions for the unobserved labels, such as ignoring the unobserved labels, assuming negative, giving them a down-weight, or starting with a random probability. Therefore, the whole labels per image, in unlabeled set, are addressed based on any of the previously mentioned assumptions during the training. Consequently, the SSPL unlabeled set's images negatively impact the performance of these models.

2.2 Semi-supervised Learning

Most of semi-supervised learning (SSL) methods on multi-class classification, while a few of them study multi-label classification [1,3,6,25,23,28,30,32,36,38,37]. Among those methods, it is usually assumed that some of the training images are fully labeled while the others are totally unlabeled, i.e., the SFL setting in Fig. 1. Most state-of-the-art methods create pseudo labels for the unlabeled data based on different approaches such as self-training-based and consistency-based approaches. Self-training-based approaches follow these steps to train the semi-supervised model: (i) train the model based on the labeled data, (ii) use the trained model to get the predictions for the unlabeled data, and (iii) apply the threshold-based strategy to the predictions to select the pseudo labels for unobserved data, which are considered only for the predictions greater than the threshold value i.e., [1,30]. Therefore, the selected pseudo labels are used in the loss function for training in the next epoch. On the other hand, the consistency-based approaches follow this strategy: (i) find the supervised loss based on the labeled data, (ii) apply the data perturbations, i.e., data augmentation and stochastic regularization, to produce two versions for the same image, and (iii) use the prediction of one image-version after applying the threshold as the pseudo label for the other image-version [3,32,36]. To be concluded, most of semi-supervised models generate the pseudo labels based on the predictions that greater than the threshold value. To make the semi-supervised models applicable for SSPL setting, the unobserved labels in the labeled set of SSPL are ignored since the labeled data of SSPL only contains one single label per image. Therefore, although the semi-supervised can be directly trained on SSPL setting, the performance of these models significantly decreased since the supervised loss function only works with a single label as discussed in subsection 4.4. While in our case, all the unobserved labels, in labeled and unlabeled sets of SSPL, initiate with unbiased probabilities to consider as initial pseudo labels and then improve their probabilities during the training using our momentum-based method. We also adapt some semi-supervised models to be applicable for multi-label classification problems, by replacing Softmax layer with Sigmoid, that are designed for multi-class classification, such as [32].

2.3 Curriculum Learning

Curriculum learning [2] has been applied to many applications related to multi-labels or multi-classes. Given the training data with fully observed labels, different iterative self-paced learning algorithms [21,19] have been developed to enhance traditional image classification, object localization, and multimedia event detection – in each iteration, easy samples are selected and the model parameters are updated accordingly. Recently, CurriculumNet [16] is proposed to learn knowledge from large-scale noisy web images by following the curriculum learning strategy. However, this strategy employs a clustering-based model to assess the complexity of the image and cannot be used for multi-label image classification [13]. Our PLMCL also follows the general idea of curriculum learning [40] by starting the learning on observed labels, and then progressively moving to the unobserved labels.

3 The PLMCL Method

Assume that the classification network is trained on the image dataset $\mathcal{I}$ for multi-label classification. Since we focus on FSPL and SSPL settings, let $\mathcal{D}$ be the subset of $\mathcal{I}$, in which each image has only one positive label and the images in $\mathcal{I} \backslash \mathcal{D}$ are totally unlabeled. Note that $\mathcal{D} \subset \mathcal{I}$ means SSPL setting and $\mathcal{D} = \mathcal{I}$ means FSPL setting. Let L be the total number of classes. For each image $i \in \mathcal{D}$, we denote $y_o^i \in \{\emptyset, 1\}^L$ as the observed label vector where each

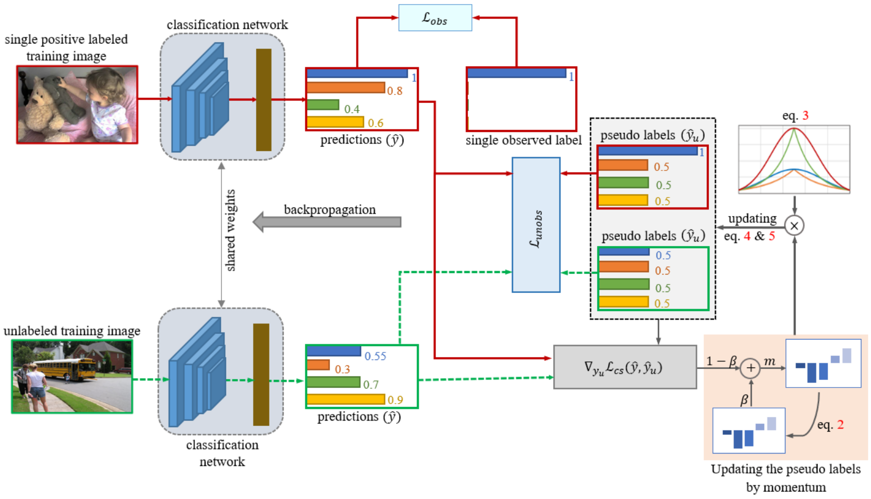

Fig. 2: An overview of the PLMCL method that consists of classification network, backbone and one-layer classifier, to generate a prediction for an input image. The classification network is end-to-end trained using the scheduled cross-entropy loss function. The pseudo labels $\hat{y}_{u,t}$ are updated based on the momentum m_t, which is the weighted sum of the last momentum m_{t-1} and the gradient of the loss between the predicted labels $\hat{y}_{t-1}$ and the pseudo labels $\hat{y}_{u,t-1}$ (Orange region). At the test, only the classification network is applied.

entry of y_o^i can be 1 (observed positive) or $\emptyset$ (unlabeled). Meantime, since the network is trained over epochs, we use $\hat{y}_{u,t}^i \in [0,1]^L$ to denote the soft pseudo label vector of of image i at epoch t [35]. Let f denote the classification network and θ_t be the network parameter obtained at epoch t. Given an image $i \in \mathcal{I}$, the predicted labels from the network is defined as $\hat{y}_t^i = f(i, \theta_t) \in [0,1]^L$. The binary cross-entropy loss function between two scalars $p, q \in [0,1]$ is defined as $\mathcal{L}(p,q) = -p\log(q) - (1-p)\log(1-q)$. Given $v \in \mathbb{R}^L$, we use $[v]_j$ to denote the jth entry of v. For notation convenience, we drop the index t if it is clear in context.

3.1 Overview

The PLMCL method is proposed for multi-label classification to deal with the case where each image in $\mathcal{D}$ only contains a single positive label. The procedure of PLMCL is shown in Fig. 2. During training, the pseudo label vector $\hat{y}_{u,0}$ of each image in $\mathcal{I}$ will be initialized with the same values of y_o for the observed labels and the unbiased probability 0.5 for the unobserved labels. Then PLMCL updates the pseudo label $\hat{y}_{u,t}$ based on the momentum m_t, which is the weighted sum of the last momentum m_{t-1} and the gradient of the loss between the predicted label vector $\hat{y}_{t-1}$ and the pseudo label vector $\hat{y}_{u,t-1}$ (Orange region in Fig. 2). It is expected that the pseudo labels are adaptively refined to provide stability for the confident labels and high momentum for the unconfident labels. The classification network is only employed during the test to predict the labels.

3.2 Updating Pseudo Labels by Momentum

Momentum Generation. When $\mathcal{D}$, the subset of the labeled images, is small, training has to start with a very limited number of observed labels, and therefore the resulting predictions will be inaccurate. In this case we cannot directly aggregate the predictions from the previous epoch to the current epoch using the off-the-shelf temporal ensembling techniques [23] to update the pseudo labels. Otherwise, the aggregation may soon arrive at a local minimum with low prediction confidence. Instead, our idea is to exploit the aggregation of the gradient (the momentum [15]) to update the pseudo label vector (Section 4.4 includes ablation study to compare these two approaches). By this way, we can help avoid undesired local minima [31].

To do so, we consider the binary cross-entropy loss function

$$\mathcal{L}_{cs}(\hat{y}_{t-1}, \hat{y}_{u,t-1}) = \frac{1}{L}\sum_{j=1}^{L} \mathcal{L}\left([\hat{y}_{t-1}]_j, [\hat{y}_{u,t-1}]_j\right) \tag{1}$$

and introduce the momentum vector m_t, which is updated based on the gradient of the loss function at epoch $t-1$, i.e., $\nabla\mathcal{L}_{cs}(\hat{y}_{t-1}, \hat{y}_{u,t-1})$:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\nabla\mathcal{L}_{cs}(\hat{y}_{t-1}, \hat{y}_{u,t-1}), \tag{2}$$

where β_1 is moving-average decay scalar and m_0 is initialized by zero. Notice that even when $\nabla\mathcal{L}_{cs} = 0$ (which means that the loss reaches its local minima), the momentum will still be updated, which can therefore avoid premature convergence. This is especially important at the early stage of training. Another observation is that, when one entry of the momentum vector consistently moves towards the same direction, the estimation of the associated pseudo label will be more confident and the variations can be well smoothed over consequence epochs which leads to a more stable training behavior.

Self-Guided Momentum Factor. Intuitively, when the confidence of the pseudo label for a class is low, we want the corresponding momentum to contribute more to the pseudo label's iteration, in order to drive the pseudo label away from its current value to avoid undesired local minima. Otherwise, we want to keep the pseudo label, in which case the momentum should contribute less. With this idea, we introduce the self-guided momentum factor $\psi(\hat{y}_{u,t}) \in \mathbb{R}^L$ at epoch t:

$$\psi(\hat{y}_{u,t}) = \alpha \exp(-\lambda|2\hat{y}_{u,t} - \mathbf{1}|^n), \tag{3}$$

where $\alpha > 0$ is the curriculum learning rate, $\lambda > 0$ and n are tuned parameters, and $\mathbf{1} \in \mathbb{R}^L$ is the vector with 1 at each entry. The exponential function exp and the absolute value $|\cdot|$ are operated element-wisely. In all our experiments, we assign n with order 2. Notice that the confidence of a pseudo label is reflected by the term $|2\hat{y}_{u,t} - \mathbf{1}|$ in the definition of ψ. For instance, if $[\hat{y}_{u,t}]_j$, the jth entry of $\hat{y}_{u,t}$, is 0.5, the related momentum factor will achieve its maximal value α, which means that our module is not confident about the pseudo label and the related momentum should contribute more in the next iteration of the pseudo label.

Otherwise, if $[\hat{y}_{u,t}]_j = 0$ or $[\hat{y}_{u,t}]_j = 1$, the related momentum factor reaches its minimum $\alpha \exp(-\lambda)$ (because $[\hat{y}_{u,t}]_j \in [0,1]$), which indicates high confidence on the current pseudo label and the momentum should contribute less so that the value of the pseudo label can be maximally kept.

With the self-guided momentum factor, we can update the pseudo label latent parameters $y_{u,t} \in \mathbb{R}^L$ by:

$$y_{u,t} = y_{u,t-1} - \psi(\hat{y}_{u,t-1}) \circ m_t \tag{4}$$

where $\circ$ means the element-wise multiplication. From this equation, we can see that the momentum m_t actually indicates a weighted difference between $y_{u,t}$ and $y_{u,t-1}$. Keeping this in mind, Equation (2) yields an interesting observation: Our approach expects not only the convergence of the pseudo labels (as it is in [23]), but also the convergence of the updating velocity of the pseudo labels. By doing so, unexpected local minima can be effectively avoided [15].

Notice that the entry of $y_{u,t}$ might be outside the interval $[0,1]$. So we need to regulate $y_{u,t}$ to obtain $\hat{y}_{u,t}$ by

$$[\hat{y}_{u,t}]_j = \sigma([y_{u,t}]_j) \tag{5}$$

for $j = 1, 2, \cdots, L$, where $\sigma : \mathbb{R} \to [0,1]$ is the sigmoid function that maps each entry of $y_{u,t}$ into $[0,1]$. In practice, the gradient in equation (2) is actually with respect to $y_{u,t-1}$, i.e., $\nabla \mathcal{L}_{cs} = \nabla_{y_{u,t-1}} \mathcal{L}_{cs}(\hat{y}_{t-1}, \hat{y}_{u,t-1})$.

3.3 Scheduled Loss Function

To learn the parameters of the primary multi-label CNN, the following loss function must be minimized

$$\mathcal{L}_{PLMCL} = \mathcal{L}_{obs} + \mathcal{L}_{unobs}, \tag{6}$$

where $\mathcal{L}_{obs}$ is the cross-entropy loss between the predicted labels and the ground truth from the observed labels plus the regularizer [10] and $\mathcal{L}_{unobs}$ is the weighted cross-entropy loss between the predicted labels and the pseudo labels for the unobserved labels

$$\mathcal{L}_{unobs} = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{U}_i} \xi([\hat{y}_{u,t}]_j, \varphi_t) \mathcal{L}\left([\hat{y}_t]_j, [\hat{y}_{u,t}]_j\right). \tag{7}$$

Here $\mathcal{U}_i$ is the set of the unobserved labels on image i and $\xi([\hat{y}_{u,t}]_j, \varphi_t)$ is a new scheduler that weights the unobserved loss, defined by

$$\xi([\hat{y}_{u,t}]_j, \varphi_t) = \beta_2 \frac{1 - \gamma \exp(-10|2[\hat{y}_{u,t}]_j - 1|)}{1 + \gamma \exp(-10|2[\hat{y}_{u,t}]_j - 1|)}, \tag{8}$$

where β_2 is the positive hyper-parameter, $\gamma = 1 - \varphi_t$, and φ_t is the current epoch t divided by the total number of epochs. This scheduler plays a critical role in balancing the contributions of unobserved labels to the total loss and

Algorithm 1 Momentum Curriculum Algorithm

```
Require: input image $i \in \mathcal{I}$;
Require: observed label $y_o$;
Require: neural network $f(i, \theta)$ with trainable parameters $\theta$ and input $i$.
Require: $[\hat{y}_u]_j = 1$ or $0$,  if labeled
    $[\hat{y}_u]_j = 0.5$,  if unlabeled
Repeat
    $\hat{y} \longleftarrow f(i, \theta)$;
    $m \longleftarrow \beta_1 m + (1 - \beta_1) \nabla_{y_u} \mathcal{L}_{cs}(\hat{y}, \hat{y}_u)$;
    $y_u \longleftarrow y_u - \psi(\hat{y}_u) m$;
    $\mathcal{L}_{PLMCL} \longleftarrow \mathcal{L}_{obs} + \mathcal{L}_{unobs}$;
    update $\theta$ via backpropagation to improve the classification model based on curriculum framework;
    $\hat{y}_u \longleftarrow \sigma(y_u)$;
until the max iteration or convergence;
Output optimal CNN parameters, $\hat{y}_u$ able to guide the prediction $\hat{y}$ to be very close from true label.
```

the potential negative impacts of false pseudo labels. When a pseudo label has low confidence level, the related loss is in appropriate, the scheduler will assign a small weight to this loss such that the total loss $\mathcal{L}_{unobs}$ will not be significantly affected. This is especially important at the beginning of the training process when the pseudo labels are still not indicative. Different from most previous schedulers relying on the training steps or amount of available training data [18], our scheduler is adaptive, counting for not only the training steps, but the validation performance during training as well. To be specific, our scheduler ξ uses the confidence $|2[\hat{y}_{u,t}]_j - 1|$ to determine the scheduler value. When $[\hat{y}_{u,t}]_j$ is close to 0.5, $|2[\hat{y}_{u,t}]_j - 1|$ will be close to 0 and therefore the value of ξ will be small. Over epochs, the pseudo labels can gradually build up their confidence and start to converge. In that case, the value of ξ will increase.

We use soft pseudo labels, instead of sharp 1 or 0, in $\mathcal{L}_{unobs}$, because the latter may make the predictions of the output layer over-confident [34]. In contrast, using the pseudo labels as the distribution scores helps smooth the predictions, which is consistent with the idea of label-smoothing regularization [34].

4 Experiments

4.1 Datasets Preparation

Datasets. The **PASCAL VOC** dataset [14] contains natural images from 20 different classes, including 5,717 training images and 5,823 images in the official validation set for testing. The **MS-COCO** dataset [24] consists of 80 classes, including 82,081 training images and 40,137 testing images. The **NUS-WIDE** dateset [9] consists of nearly 150K color images with various resolutions for training and 60.2K for testing. This dataset is associated with 81 classes.

Data Preparation. The SSPL setting annotates certain percent of the training images (20% to 80%) with only a single positive label and the rest of the images are totally unlabeled. The FSPL setting has 100% of the training images annotated with only one single positive label. To do so, we follow [10] by assigning randomly only one positive label for each image in the training set.

Implementation Details. For a fair comparison, we use ResNet-50 to initialize the weights of the backbone architecture for all models. The learning rate is 10^{-5} to train the VOC and COCO datasets and 10^{-4} to train the NUS dataset. The batch size is chosen as 8 for the VOC dataset, and 16 for the COCO and NUS datasets. Training is performed in two ways: end-to-end and linear-init. End-to-end training simultaneously updates the parameters of the backbone and the classifier for 10 epochs, while linear-init training first fixes the parameters of the backbone and trains the classifier for 25 epochs, and then fine-tunes the parameters of the classifier and the backbone for 5 epochs [10]. The best mAP on the validation set is reported for all experiments. It is worth mentioning that all experiments on our model share the same set of hyper-parameters. The mean average precision (mAP) is applied to evaluate the performance of different approaches for multi-label classification in our paper, similar to [10,17].

Table 1: Quantitative results (mAP) of multi-label image classification on different subsets of the single observed label (SSPL) setting on COCO, VOC, and NUS datasets. Bold represents the highest mAP and underline represents the second-best.

Losses	**COCO dataset**							
	end-to-end				linear-init			
	20%	40%	60%	80%	20%	40%	60%	80%
$\mathcal{L}_{AN}$	46.3	53.8	59.5	62.4	53.4	61.1	63.8	65.6
$\mathcal{L}_{AN-LS}$	48.9	57.9	62.3	$\underline{65.5}$	58.2	62.8	$\underline{67.6}$	68.5
$\mathcal{L}_{WAN}$	$\underline{57.0}$	$\underline{60.9}$	$\underline{63.5}$	64.5	60.2	62.4	64.1	66.1
$\mathcal{L}_{EPR}$	52.7	58.4	61.5	62.6	61.0	63.6	65.3	66.6
$\mathcal{L}_{ROLE}$	47.3	57.6	62.7	65.2	$\underline{61.9}$	$\underline{65.8}$	67.4	$\underline{68.6}$
$\mathcal{L}_{PLMCL}$	**61.0**	**65.1**	**68.4**	**70.4**	**64.8**	**67.8**	**69.2**	**71.5**
Losses	**VOC dataset**							
	end-to-end				linear-init			
	20%	40%	60%	80%	20%	40%	60%	80%
$\mathcal{L}_{AN}$	51.3	71.7	80.2	82.8	78.6	81.4	83.7	84.9
$\mathcal{L}_{AN-LS}$	70.0	79.0	85.0	86.1	71.5	81.4	84.9	86.4
$\mathcal{L}_{WAN}$	$\underline{76.4}$	$\underline{82.5}$	85.1	85.6	79.4	83.6	$\underline{85.9}$	86.9
$\mathcal{L}_{EPR}$	75.5	81.0	83.9	84.5	$\underline{82.6}$	$\underline{84.9}$	$\underline{85.9}$	86.9
$\mathcal{L}_{ROLE}$	66.9	80.9	$\underline{85.9}$	$\underline{86.8}$	73.9	82.3	85.7	$\underline{87.5}$
$\mathcal{L}_{PLMCL}$	**79.9**	**84.6**	**87.6**	**88.4**	**83.1**	**85.4**	**87.8**	**87.7**
Losses	**NUS dataset**							
	end-to-end				linear-init			
	20%	40%	60%	80%	20%	40%	60%	80%
$\mathcal{L}_{AN}$	28.5	35.2	38.9	40.8	35.2	42.8	44.8	46.5
$\mathcal{L}_{AN-LS}$	27.8	36.0	39.1	41.4	39.1	44.4	$\underline{47.2}$	48.0
$\mathcal{L}_{WAN}$	$\underline{37.6}$	$\underline{41.3}$	$\underline{43.7}$	$\underline{44.8}$	41.4	44.5	45.7	46.2
$\mathcal{L}_{EPR}$	34.9	39.5	42.3	44.2	$\underline{43.4}$	$\underline{46.0}$	$\underline{47.2}$	$\underline{48.2}$
$\mathcal{L}_{ROLE}$	27.0	32.2	37.1	39.7	36.3	42.8	46.4	47.5
$\mathcal{L}_{PLMCL}$	**37.9**	**42.5**	**44.8**	**46.6**	**46.2**	**49.4**	**50.3**	**50.7**

4.2 Subset Single Positive Label (SSPL)

This subsection compares the performance of our model $\mathcal{L}_{PLMCL}$ with other state-of-the-art approaches that address the unobserved labels in multi-label classification under SSPL settings. The comparison results are listed in Table 1. We observe that $\mathcal{L}_{AN}$ [22] has the worst performance since it assumes that all the unobserved labels are negatives. This assumption adds more noisy labels (false negatives) into the training process. To overcome this negative influence, $\mathcal{L}_{AN-LS}$ is introduced using label smoothing on $\mathcal{L}_{AN}$. Following [10], we use

Table 2: Quantitative results (mAP) of multi-label image classification for FSPL setting on three different datasets with end-to-end and linear-init training. Bold represents the highest mAP and underline represents the second-best.

		COCO dataset		**VOC dataset**		**NUS dataset**	
Supervised by	Losses	end-to-end	linear-init	end-to-end	linear-init	end-to-end	linear-init
Fully Labeled	$\mathcal{L}_{BCE}$	75.5	76.8	89.1	90.1	52.6	54.7
	$\mathcal{L}_{BCE-LS}$	76.8	78.8	90	90.9	53.5	54.2
1 Positive (FSPL Setting)	$\mathcal{L}_{AN}$	64.1	67.3	85.1	87.7	42	46.8
	$\mathcal{L}_{AN-LS}$	$\underline{66.9}$	$\underline{69.2}$	86.7	86.5	44.9	50.5
	$\mathcal{L}_{WAN}$	64.8	67.8	86.5	87.1	$\underline{46.3}$	47.5
	$\mathcal{L}_{EPR}$	63.3	67.5	85.5	85.6	46.0	48.9
	$\mathcal{L}_{ROLE}$	66.3	69.0	$\underline{87.9}$	$\underline{88.2}$	43.1	$\underline{51.0}$
	$\mathcal{L}_{PLMCL}$	**71.1**	**72.0**	**89.0**	**88.7**	**48.6**	**51.5**

$\mathcal{L}_{AN-LS}$ as a baseline. To go one step further in mitigating the deteriorated effect of noisy labels, some models tend to down-weight the term related to negative labels in the loss function $\mathcal{L}_{WAN}$ [26]. Consequently, $\mathcal{L}_{WAN}$ improves the mAPs on different datasets as compared to $\mathcal{L}_{AN}$ and $\mathcal{L}_{AN-LS}$. Finally, expected positive regularization $\mathcal{L}_{EPR}$ [10] and online estimation of unobserved labels $\mathcal{L}_{ROLE}$ are introduced in [10] to deal with the single positive label, which is considered an extreme case of unobserved labels, in multi-label classification. We notice that $\mathcal{L}_{EPR}$ [10] provides stable performance in different settings since $\mathcal{L}_{EPR}$ ignores the unobserved terms and focuses on finding the positive labels according to its regularizer. Also notice that the performance of $\mathcal{L}_{ROLE}$ degrades a lot under 20% SSPL in end-to-end setting since its classifier and label estimator are randomly initialized [10]. Compared with these methods, our proposed $\mathcal{L}_{PLMCL}$, based on the momentum, is superior in the three different datasets.

4.3 Full-set Single Positive Label (FSPL)

This subsection compares different models under FSPL setting. Table 2 reports the mAPs over three different datasets under two training settings: end-to-end and linear-init. Besides FSPL, we also include in Table 2 the mAPs of two models ($\mathcal{L}_{BCE}$ and $\mathcal{L}_{BCE-LS}$) based on full-set fully labeled (FFL) setting [10], as references to examine the performance of our model under FSPL setting. Compared with the FFL setting using 100% of the labels, the FSPL setting uses only around 1.2% of the total number of labels. It is worth mentioning that the mAPs of our model $\mathcal{L}_{PLMCL}$ under FSPL remain at a comparable level. For instance, the mAP of our model on the NUS dataset is 48.6%, which is only 4.0% lower than the FFL setting $\mathcal{L}_{BCE}$. Similar conclusions can be drawn in COCO and VOC datasets. When comparing the approaches under FSPL setting (the bottom part of Table 2), we observe that our model $\mathcal{L}_{PLMCL}$ always achieves the best performance on different datasets.

Backbone ResNet-101 . Besides ResNet-50, we also conducted experiments using a different backbone, ResNet-101, in the end-to-end setting on two datasets under different SSPL settings. The results are reported in Table 3. Our model

Table 3: Quantitative results (mAP) of multi-label image classification on different subsets of the single observed label (SSPL) setting and FSPL (100%) setting on two different datasets with end-to-end learning using the Resnet-101 backbone. Bold represents the highest mAP and underline represents the second-best.

Models	**COCO Dataset**					**VOC Dataset**				
	SSPL				FSPL	SSPL				FSPL
	20%	40%	60%	80%	100%	20%	40%	60%	80%	100%
$\mathcal{L}_{AN}$	42.0	52.4	57.9	62.3	63.9	52.9	73.6	82.1	85.1	86.1
$\mathcal{L}_{AN-LS}$	49.2	57.5	61.8	65.6	66.9	67.1	80.6	85.8	86.8	88.3
$\mathcal{L}_{WAN}$	$\underline{57.3}$	$\underline{61.8}$	$\underline{64.9}$	$\underline{66.0}$	66.7	$\underline{76.1}$	$\underline{83.1}$	85.9	86.5	87.2
$\mathcal{L}_{EPR}$	51.8	57.6	60.9	62.8	63.8	75.1	80.3	85.0	84.6	86.7
$\mathcal{L}_{ROLE}$	46.2	58.3	62.8	65.5	$\underline{67.3}$	66.7	82.3	$\underline{86.2}$	$\underline{87.4}$	$\underline{88.8}$
$\mathcal{L}_{PLMCL}$ **(ours)**	**61.4**	**67.3**	**69.5**	**71.5**	**72.2**	**82.0**	**86.2**	**88.4**	**88.8**	**89.7**

$\mathcal{L}_{PLMCL}$ achieves the hightest mAP scores under all settings. It is worth mentioning that our model $\mathcal{L}_{PLMCL}$ at 60% of SSPL setting achieves higher mAP compared to $\mathcal{L}_{ROLE}$ under FSPL setting (100%) in COCO dataset, which demonstrates the efficiency of our model in saving labeling costs.

4.4 Ablation Study.

PLMCL Modules. Our ablation analysis highlights the importance of each component in our method: the momentum m_t, the scheduler ξ, and the self-guided factor ψ. The results are reported in Table 4. We start with the basic variation of the model which uses the cross-entropy loss on the observed labels. In this case, we assume that all the unobserved labels are negative, since ignoring the unobserved labels may lead to overfitting. When adding components into the model, this assumption can be removed since we use the pseudo labels to deal with the unobserved labels. We first add the momentum generation module to update the soft pseudo labels which improve the mAP for different settings and datasets (Table 4, second row). Adding the confidence-aware scheduler and the self-guided factor further enhances in the overall performance, since they work together to make the training adaptive with respect to the confidence of the pseudo labels (Table 4, last two rows).

Table 4: Ablation study for all proposed model components on SSPL (20%) and FSPL with linear-init training setting on COCO dataset.

Hyper Parameters			COCO Dataset (mAP %)	
m_t	ξ	ψ	20%	100%
			59.0	67.3
✓			62.0	69.9
✓	✓		63.1	70.4
✓	✓	✓	64.8	72.0

Hyper Parameters. The influence of the hyper-parameters is studied on the VOC validation set under the 40% SSPL setting. We run experiments with different hyper-parameters. In all of these experiments, our model remains stable and reliable. Table 5 shows the variations in mAP according to the changes of hyper-parameters β_1, β_2, and λ. Recall that the hyper-parameter β_1 is used to balance the previous network knowledge and the current decision of the network.

Table 5: mAP score (%) of our method as a function of β_1, β_2, λ in subset single positive setting SSPL at 40% using end-to-end training setting on the VOC validation set.

β_1	0.1	0.5	0.7	0.9
mAP (%) ($\beta_2 = 0.6$)	84.4	84.5	**84.6**	84.0

β_2	0.1	0.4	0.6	0.9
mAP (%) ($\beta_1 = 0.7$)	83.5	84.5	**84.6**	84.5

λ	1	2	3	4
mAP (%) ($\beta_1 = 0.7$, $\beta_2 = 0.6$)	83.0	83.2	83.9	**84.6**

We observed that the mAP score improves when the balance is achieved (β_1 is close to 0.5) and decreases in the extreme cases ($\beta_1 = 0.1$ emphasizing the current decision of the network and $\beta_1 = 0.9$ emphasizing the previous network knowledge). λ demonstrates the effectiveness of the self-guided factor ψ that controls the amount of momentum contributing to the update of pseudo levels. The results in Table 5 indicates that the mAP score increases as the λ increases. When increasing the scheduler parameter β_2, the mAP score almost remains the same level. Overall, our model is not very sensitive to the variations in these hyper-parameters.

Comparison with Semi-supervised Models. We compare our PLMCL with the existing semi-supervised models such as FixMatch [32] and UPS [30] on COCO dataset. Fig. 3 shows the comparison of the mAP results under different SSPL settings. Notice that PLMCL always outperforms the other two methods. This is because FixMatch and UPS are trained based on the supervised loss over the set of observed labels while ignoring the unobserved labels of the labeled set as explained in subsection 2.2. Therefore, there is a lack of continuity in pseudo label updating. On the contrary, PLMCL can guarantee such continuity and capture the temporal information during training. So it is not surprising that PLMCL achieves the best mAP score.

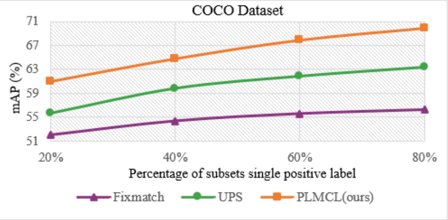

Fig. 3: Ablation study on the difference between using our method PLMCL and semi-supervised models.

5 Conclusions

This paper presents a momentum-based curriculum learning method for multi-label image classification under partial-label settings. In PLMCL, soft pseudo labels are generated per image and then updated based on the momentum and the confidence levels. Through extensive experiments on three different partial-label settings, including our proposed new SSPL setting, we demonstrated that our PLMCL outperforms the state-of-the-art methods, especially when training on fewer observed labels.

Acknowledgements. The authors gratefully acknowledge the partial financial support of the National Science Foundation (1830512).

References

1. Arazo, E., Ortego, D., Albert, P., O'Connor, N.E., McGuinness, K.: Pseudo-labeling and confirmation bias in deep semi-supervised learning. In: 2020 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2020)
2. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: 26th annual International Conference on Machine Learning (ICML). pp. 41–48 (2009)
3. Berthelot, D., Carlini, N., Cubuk, E.D., Kurakin, A., Sohn, K., Zhang, H., Raffel, C.: Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. arXiv preprint arXiv:1911.09785 (2019)
4. Bucak, S.S., Jin, R., Jain, A.K.: Multi-label learning with incomplete class assignments. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2801–2808. IEEE (2011)
5. Cabral, R.S., Torre, F., Costeira, J.P., Bernardino, A.: Matrix completion for multi-label image classification. In: Advances in Neural Information Processing Systems. pp. 190–198 (2011)
6. Chapelle, O., Scholkopf, B., Zien, A.: Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. IEEE Transactions on Neural Networks **20**(3), 542–542 (2009)
7. Chen, M., Zheng, A., Weinberger, K.: Fast image tagging. In: International Conference on Machine Learning (ICML). pp. 1274–1282. PMLR (2013)
8. Chu, H.M., Yeh, C.K., Wang, Y.C.F.: Deep generative models for weakly-supervised multi-label classification. In: European Conference on Computer Vision (ECCV). pp. 400–415 (2018)
9. Chua, T.S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.: Nus-wide: a real-world web image database from national university of singapore. In: ACM International Conference on Image and Video Retrieval. pp. 1–9 (2009)
10. Cole, E., Mac Aodha, O., Lorieul, T., Perona, P., Morris, D., Jojic, N.: Multi-label learning from single positive labels. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 933–942 (2021)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 248–255. Ieee (2009)
12. Deng, J., Russakovsky, O., Krause, J., Bernstein, M.S., Berg, A., Fei-Fei, L.: Scalable multi-label annotation. In: SIGCHI Conference on Human Factors in Computing Systems. pp. 3099–3102 (2014)
13. Durand, T., Mehrasa, N., Mori, G.: Learning a deep convnet for multi-label classification with partial labels. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 647–657 (2019)
14. Everingham, M., Winn, J.: The pascal visual object classes challenge 2012 (voc2012) development kit. Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep **8**, 5 (2011)
15. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT press (2016)
16. Guo, S., Huang, W., Zhang, H., Zhuang, C., Dong, D., Scott, M.R., Huang, D.: Curriculumnet: Weakly supervised learning from large-scale web images. In: European Conference on Computer Vision (ECCV). pp. 135–150 (2018)
17. Huynh, D., Elhamifar, E.: Interactive multi-label cnn learning with partial labels. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9423–9432 (2020)

18. Jean, S., Firat, O., Johnson, M.: Adaptive scheduling for multi-task learning. arXiv preprint arXiv:1909.06434 (2019)
19. Jiang, L., Meng, D., Zhao, Q., Shan, S., Hauptmann, A.G.: Self-paced curriculum learning. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
20. Kapoor, A., Viswanathan, R., Jain, P.: Multilabel classification using bayesian compressed sensing. Advances in Neural Information Processing Systems **25**, 2645–2653 (2012)
21. Kumar, M., Packer, B., Koller, D.: Self-paced learning for latent variable models. Advances in neural information processing systems **23**, 1189–1197 (2010)
22. Kundu, K., Tighe, J.: Exploiting weakly supervised visual patterns to learn from partial annotations. Advances in Neural Information Processing Systems **33**, 561–572 (2020)
23. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. arXiv preprint arXiv:1610.02242 (2016)
24. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision (ECCV). pp. 740–755. Springer (2014)
25. Liu, Y., Jin, R., Yang, L.: Semi-supervised multi-label learning by constrained non-negative matrix factorization. In: AAAI. vol. 6, pp. 421–426 (2006)
26. Mac Aodha, O., Cole, E., Perona, P.: Presence-only geographical priors for fine-grained image classification. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9596–9606 (2019)
27. Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., Van Der Maaten, L.: Exploring the limits of weakly supervised pretraining. In: European Conference on Computer Vision (ECCV). pp. 181–196 (2018)
28. Niu, X., Han, H., Shan, S., Chen, X.: Multi-label co-regularization for semi-supervised facial action unit recognition. arXiv preprint arXiv:1910.11012 (2019)
29. Pineda, L., Salvador, A., Drozdzal, M., Romero, A.: Elucidating image-to-set prediction: An analysis of models, losses and datasets. CoRR (2019)
30. Rizve, M.N., Duarte, K., Rawat, Y.S., Shah, M.: In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In: International Conference on Learning Representations (2021)
31. Sariyildiz, M.B., Cinbis, R.G.: Gradient matching generative networks for zero-shot learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2168–2178 (2019)
32. Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C.A., Cubuk, E.D., Kurakin, A., Li, C.L.: Fixmatch: Simplifying semi-supervised learning with consistency and confidence. Advances in Neural Information Processing Systems **33**, 596–608 (2020)
33. Sun, C., Shrivastava, A., Singh, S., Gupta, A.: Revisiting unreasonable effectiveness of data in deep learning era. In: IEEE International Conference on Computer Vision (ICCV). pp. 843–852 (2017)
34. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2818–2826 (2016)
35. Tanaka, D., Ikami, D., Yamasaki, T., Aizawa, K.: Joint optimization framework for learning with noisy labels. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5552–5560 (2018)
36. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. Advances in neural information processing systems **30** (2017)

37. Wang, B., Tu, Z., Tsotsos, J.K.: Dynamic label propagation for semi-supervised multi-class multi-label classification. In: IEEE International Conference on Computer Vision (ICCV). pp. 425–432 (2013)
38. Wang, L., Ding, Z., Fu, Y.: Adaptive graph guided embedding for multi-label annotation. In: IJCAI (2018)
39. Wang, Q., Shen, B., Wang, S., Li, L., Si, L.: Binary codes embedding for fast image tagging with incomplete labels. In: European Conference on Computer Vision (ECCV). pp. 425–439. Springer (2014)
40. Wang, X., Chen, Y., Zhu, W.: A survey on curriculum learning. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
41. Wu, B., Lyu, S., Ghanem, B.: Ml-mg: Multi-label learning with missing labels using a mixed graph. In: IEEE International Conference on Computer Vision (ICCV). pp. 4157–4165 (2015)
42. Xu, M., Jin, R., Zhou, Z.H.: Speedup matrix completion with side information: Application to multi-label learning. In: Advances in Neural Information Processing Systems. pp. 2301–2309 (2013)
43. Yang, H., Zhou, J.T., Cai, J.: Improving multi-label learning with missing labels by structured semantic correlations. In: European Conference on Computer Vision (ECCV). pp. 835–851. Springer (2016)
44. Yu, H.F., Jain, P., Kar, P., Dhillon, I.: Large-scale multi-label learning with missing labels. In: International Conference on Machine Learning (ICML). pp. 593–601. PMLR (2014)