Large Loss Matters in Weakly Supervised Multi-Label Classification

Youngwook Kim[1*] Jae Myung Kim[2*] Zeynep Akata[2,3,4] Jungwoo Lee[1,5]

[1]Seoul National University [2]University of Tübingen
[3]Max Planck Institute for Intelligent Systems [4]Max Planck Institute for Informatics [5]HodooAI Lab

Abstract

Weakly supervised multi-label classification (WSML) task, which is to learn a multi-label classification using partially observed labels per image, is becoming increasingly important due to its huge annotation cost. In this work, we first regard unobserved labels as negative labels, casting the WSML task into noisy multi-label classification. From this point of view, we empirically observe that memorization effect, which was first discovered in a noisy multi-class setting, also occurs in a multi-label setting. That is, the model first learns the representation of clean labels, and then starts memorizing noisy labels. Based on this finding, we propose novel methods for WSML which reject or correct the large loss samples to prevent model from memorizing the noisy label. Without heavy and complex components, our proposed methods outperform previous state-of-the-art WSML methods on several partial label settings including Pascal VOC 2012, MS COCO, NUSWIDE, CUB, and OpenImages V3 datasets. Various analysis also show that our methodology actually works well, validating that treating large loss properly matters in a weakly supervised multi-label classification. Our code is available at https://github.com/snucml/LargeLossMatters.

1. Introduction

Multi-label classification aims to find all existing objects or attributes in a single image. It is gaining attention since the real world is made up of a scene with multiple objects in it [28, 35]. Moreover, some of the single-label datasets, also called multi-class datasets, actually have images containing multiple objects [33, 56]. However, the multi-label classification task has some fundamental difficulties in making a dataset because it requires annotators to label all categories' existence/absence for every image. As the number of categories and images in the dataset increase, annotation cost becomes tremendous [19].

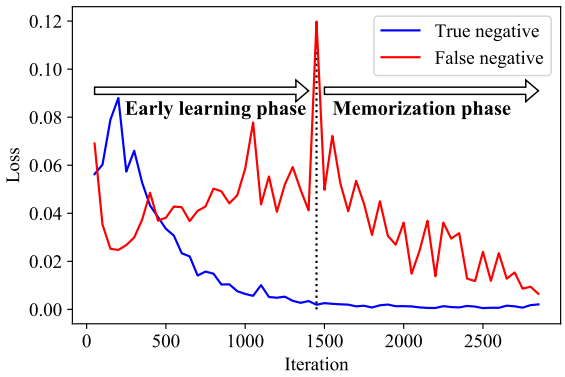

Figure 1. **Memorization in WSML.** When training ResNet-50 model on PASCAL VOC dataset with partial label, we set all unobserved labels as negative. These labels are composed of true negative and false negative. We observe that the model first fits into true negative label (learning), and then fits into false negative (memorization).

To alleviate these issues, weakly supervised learning approach in multi-label classification task (WSML) has been taken into consideration [2, 18, 36, 50]. In a WSML setting, labels are given as a form of partial label, which means only a small amount of categories is annotated per image. This setting reflects the recently released large-scale multi-label datasets [12, 19] which provide only partial label. Thus, it is becoming increasingly important to develop learning strategies with partial labels.

There are two naive approaches to train the model with partial labels. One is to train the model with observed labels only, ignoring the unobserved labels. The other is to assume all unobserved labels are negative and incorporate them into training because majorities of labels are negative in a multi-label setting [32]. As the second one has a limitation that this assumption produces some noise in a label which hampers the model learning, previous works [7, 9, 16, 21] mostly follow the first approach and try to explore the cue of unobserved labels using various techniques such as bootstrapping or regularization. However, these approaches include

*Equal contribution.

heavy computation or complex optimization pipeline.

We hypothesize that if label noise can be handled properly, the second approach could be a good starting point because it has the advantage of incorporating many true negative labels into model training. Therefore, we try to look at the WSML problem from the perspective of noisy label learning.

Our key observation is about the memorization effect [1] in a noisy label learning literature. It is known that when training a model with a noisy label, the model fits into clean labels first and then starts memorizing noisy labels. Although previous work showed the memorization effect only in a noisy *multi-class* classification scenario, we found for the first time that this same effect also happens in a noisy *multi-label* classification scenario. As shown in Figure 1, during training, the loss value from the clean label (true negative) decreases from the beginning while the loss from the noisy label (false negative) decreases from the middle.

Based on this finding, we borrow the idea from noisy multi-class literature [13, 17, 23] which selectively trains the model with samples having small loss and adapt this idea into a multi-label scenario. Specifically, by assigning the unknown labels as negative in a WSML setting, label noise appears in the form of false negative. Then we develop the three different schemes to prevent false negative labels from being memorized into the multi-label classification model by rejecting or correcting large loss samples during training.

Our method is light and simple, yet effective. It involves negligible computation overhead and does not require complex optimization for model training. Nonetheless, our method surpasses the weakly supervised multi-label classification performance compared to the state-of-the-art methods in Pascal VOC 2012 [10], MS COCO [24], NUSWIDE [6], CUB [42], and OpenImages V3 [19] datasets. Moreover, while some existing methods are only effective in specific partial label setting [7, 9, 16], our method is broadly applicable in both artificially created and real partial label datasets. Finally, we provide some analysis about the reason why our methods work well from various perspectives.

To sum up, our contributions are as follows;

1) We empirically show for the first time that the memorization effect occurs during noisy multi-label classification.

2) We propose a novel scheme for weakly supervised multi-label classification that explicitly utilizes a learning technique with noisy label.

3) Although light and simple, our proposed method achieves state-of-the-art classification performance on various partial label datasets.

2. Related Works

Multi-label classification. The main research trend of this field has been modeling correlations between labels [15, 31, 38, 55] because multiple objects can appear simultaneously in a multi-label setting. Recently this modeling was realized through graph neural networks [4, 5, 53], recurrent models [43, 52], or transformer encoder structure [22]. Recent research trends also include solving imbalance issues in multi-label dataset such as long-tail class distribution [11, 48] or positive-negative label imbalance [32].

Weakly supervised multi-label classification. Due to the annotation issue, weakly supervised learning of multi-label classification has been another important study. There are several approaches to train the model using partially annotated labels: regarding missing labels as negative [2, 3, 36, 44], predicting the missing labels via label correlation modeling [8, 47, 49, 50] or probabilistic model [18, 41]. Note that these methods use traditional optimization and they are not scalable to training deep neural networks.

[9] is the first work to train a deep neural network using partial label. It adopts a curriculum learning approach to label some unannotated easy samples using its model prediction. However, its initial model trained only on a partial label has a weak representation, which may lead to wrong labelling. [16, 21] models label similarity and image similarity to predict unobserved labels from other semantically similar images' features or observed labels. Recently, [7] suggested learning with only one positive label per image, which is a subset of partial label scenario. It also proposed a regularization scheme using an average number of positive labels in a dataset and alternate optimization of classifier and unobserved label estimator. However, they require complex optimization pipeline or heavy computation cost. Our method takes a different route with previous method by casting WSML into noisy multi-label classification. Note that few studies have been done in this route except for applying label smoothing [7, 21].

Noisy multi-class classification. In label noise literature, there are two major branches: one is sample selection and the other is label correction. Sample selection approach starts from the finding of [1] and tries to select only clean samples to train the model in the presence of noisy labels. The criterion of clean samples can be small-loss [13, 17, 23, 46], consistent prediction with running average of previous predictions [25, 29], low divergence between prediction and label [51]. Label correction approach tries to update the noisy label instead of viewing it as a fixed one. There are approaches for updating label into softmax-activated prediction [39], optimizing label via backprop [54], using adaptive target during training [45]. [26] showed that label smoothing can be also viewed as one of the approaches in label correction. There is also a hybrid method [34] which takes advantage of both sample selection and label correction. Our method borrows the idea of sample selection and label correction to cope with label noise in a WSML setting. However, since the noise type is different between multi-

class and multi-label, we propose a method specialized in a multi-label setting.

3. Approach

In this section, we start with the definition of assume negative (AN) in weakly supervised multi-label setting (WSML) in §3.1. Within this setting, we show in §3.2 that the model first learns features of true positive and true negative labels, and then starts memorizing false negative labels. Based on this finding, we propose three methods in §3.3, that is to modify the large loss samples during training which is likely to be from false negative labels.

3.1. Target with Assume Negative

Let us define an input $\boldsymbol{x} \in \mathcal{X}$ and a target $\boldsymbol{y} \in \mathcal{Y}$ where $\mathcal{X}$ and $\mathcal{Y}$ compose a dataset $\mathcal{D}$. In a weakly supervised multi-label learning for image classification task, $\mathcal{X}$ is an image set and $\mathcal{Y} = \{0, 1, u\}^K$ where u is an annotation of 'unknown', i.e. unobserved label, and K is the number of categories. For the target $\boldsymbol{y}$, let $\mathcal{S}^p = \{i | y_i = 1\}$, $\mathcal{S}^n = \{i | y_i = 0\}$, and $\mathcal{S}^u = \{i | y_i = u\}$. In a partial label setting, small amount of labels are known, thus $|\mathcal{S}^p| + |\mathcal{S}^n| < K$. We start our method with Assume Negative (AN) where all the unknown labels are regarded as negative. We call this modified target as $\boldsymbol{y}^{AN}$,

$$y_i^{AN} = \begin{cases} 1, & i \in \mathcal{S}^p \\ 0, & i \in \mathcal{S}^n \cup \mathcal{S}^u \end{cases}, \tag{1}$$

and the set of all $\boldsymbol{y}^{AN}$ as $\mathcal{Y}^{AN}$. $\{y_i^{AN} | i \in \mathcal{S}^p\}$ and $\{y_i^{AN} | i \in \mathcal{S}^n\}$ are the set where each element is true positive and true negative, respectively. $\{y_i^{AN} | i \in \mathcal{S}^u\}$ contains both true negative and false negative. The naive way of training the model f with the dataset $\mathcal{D}' = (\mathcal{X}, \mathcal{Y}^{AN})$ is to minimize the loss function L,

$$L = \frac{1}{|\mathcal{D}'|} \sum_{(\boldsymbol{x}, \boldsymbol{y}^{AN}) \in \mathcal{D}'} \frac{1}{K} \sum_{i=1}^{K} \text{BCELoss}\left(f(\boldsymbol{x})_i, y_i^{AN}\right), \tag{2}$$

where $f(\cdot) \in [0, 1]^K$ and $\text{BCELoss}(\cdot, \cdot)$ is the binary cross entropy loss between the function output and the target. We call this naive method as Naive AN.

3.2. Memorization in WSML

Let us first revisit the memorization effect in a noisy multi-class learning [1]. In the noisy multi-class setting, each data in a dataset is composed of an input and a target where the target is a single category with some of it annotated wrong. For clean labels, the annotated single category is true while for noisy labels the annotated category is false. When a model is trained with the dataset that contains both clean labels and noisy labels, the model first learns features of data with clean labels and then starts to memorize the data with noisy labels. This is in line with the other observation where the model first learns easy patterns and then learns more difficult patterns [40].

Highest loss phase	Pascal VOC (%)			MS COCO (%)		
	TP	TN	FN	TP	TN	FN
Warmup	**88.3**	**90.7**	23.8	**64.0**	**82.6**	17.3
Regular	11.7	9.3	**72.2**	36.0	17.4	**82.7**

Table 1. **Distribution of the highest loss occurrence.** For each label, we first draw the loss plot in the training process. We then record whether the highest loss occurred in the warmup phase (epoch 1) or in the regular phase (after epoch 1). TP, TN, FN refers to true positive, true negative, and false negative, respectively.

We observe that a similar memorization effect occurs in WSML when the model is trained with the dataset with AN target. To confirm this, we make the following experimental setting. We convert Pascal VOC 2012 [10] dataset into partial label one by randomly remaining only one positive label for each image and regard other labels as unknown (dataset $\mathcal{D}$). These unknown labels are then assumed as negative (dataset $\mathcal{D}'$). We train ResNet-50 [14] model with $\mathcal{D}'$ using the loss function L in Equation 2. We look at the trend of loss value corresponding to each label y_i^{AN} in a training dataset while the model is trained. A single example for true negative label and false negative label is shown in Figure 1. For a true negative label, the corresponding loss value keeps decreasing as the number of iteration increases (blue line). Meanwhile, the loss of a false negative label slightly increases in the initial learning phase, and then reaches the highest in the middle phase followed by decreasing to reach near 0 at the end (red line). This implies that the model starts to memorize the wrong label from the middle phase.

To see if this phenomenon constantly occurs across all the labels in a training dataset, we conduct the following experiment. For every label, we track the loss value on each training epoch. Then we count the number of labels having the largest loss in the first epoch. We perform this experiments on partially labeled Pascal VOC 2012 [10] and MS COCO dataset [24] with AN target and ResNet-50. The results are shown in Table 1. Most of true positive and true negative samples have a highest loss in the first epoch (warmup phase), whereas false negatives usually show a highest loss after the first epoch (regular phase). These results indicate that the model learns features from the data corresponding to true positive and true negative labels in the initial phase, while memorization of false negative labels generally starts in the middle of the training phase.

3.3. Method: Large Loss Modification

In this section, we propose novel methods for WSML motivated from the ideas of noisy multi-class learning [13, 17, 23] which ignores the large loss during training the

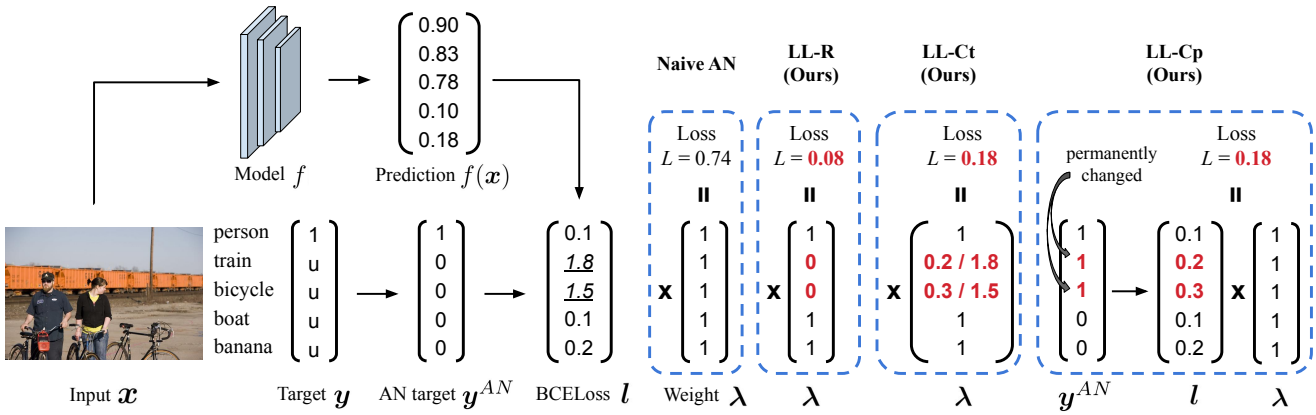

Figure 2. **Overall pipeline of our proposed methods.** We propose three different ways of dealing false negative labels in AN target y^{AN} which cause large loss. While Naive AN baseline takes average over all elements in BCELoss $\boldsymbol{l}$, our methods control the weight λ to reject or correct the false negative labels (LL-R or LL-Ct), or directly change the label from negative to positive (LL-Cp). Note that 'u' in target y means its label is unobserved.

model. Remind that in WSML with AN target, the model starts memorizing the false negative label in the middle of the training with having a large loss at that time. While we can only observe that the label in the set $\{y_i^{AN} | i \in \mathcal{S}^u\}$ is negative and cannot explicitly discriminate whether it is false or true, we are able to implicitly distinguish between them. It is because the loss from false negative is likely to be larger than the loss from true negative before memorization starts. Therefore, we manipulate the label in the set $\{y_i^{AN} | i \in \mathcal{S}^u\}$ that corresponds to the large loss value during the training process to prevent the model from memorizing false negative labels. We do not manipulate the known true labels, i.e. $\{y_i^{AN} | i \in \mathcal{S}^p \cup \mathcal{S}^n\}$, since they are all clean labels. Instead of using Equation 2 as a loss function, we further introduce the weight term λ_i in the loss function,

$$L = \frac{1}{|\mathcal{D}'|} \sum_{(\boldsymbol{x}, \boldsymbol{y}^{AN}) \in \mathcal{D}'} \frac{1}{K} \sum_{i=1}^{K} l_i \times \lambda_i \,. \tag{3}$$

We define $l_i = \mathrm{BCELoss}\,(f(\boldsymbol{x})_i, y_i^{AN})$ where arguments of function l_i, that are $f(\boldsymbol{x})$ and $\boldsymbol{y}^{AN}$, are omitted for convenience. The term λ_i is defined as a function, $\lambda_i = \lambda(f(\boldsymbol{x})_i, y_i^{AN})$, where arguments are also omitted for convenience. λ_i is the weighted value for how much the loss l_i should be considered in the loss function L in Equation 3. Intuitively, λ_i should be small when $i \in \mathcal{S}^u$ and the loss l_i has high value in the middle of the training, that is, to ignore that loss since it is likely to be the loss from a false negative sample. We set $\lambda_i = 1$ when $i \in \mathcal{S}^p \cup \mathcal{S}^n$ since the label y_i^{AN} from these indices is a clean label. We present three different schemes of offering the weight λ_i for $i \in \mathcal{S}^u$. The schematic description is shown in Figure 2.

Large loss rejection. One way of dealing with large loss sample is to reject it by setting $\lambda_i = 0$. In a noisy multi-class task, B. Han et al. [13] propose a method of gradually increasing the rejection rate during the training process. We set the function λ_i similarly,

$$\lambda_i = \begin{cases} 0, & i \in \mathcal{S}^u \text{ and } l_i > R(t) \\ 1, & \text{otherwise}\,, \end{cases} \tag{4}$$

where t is the number of current epochs in the training process and $R(t)$ is the loss value that has $[(t-1) \cdot \Delta_{rel}]\%$ largest value in the loss set $\{l_i \,|\, (\boldsymbol{x}, \boldsymbol{y}^{AN}) \in \mathcal{D}', i \in \mathcal{S}^u\}$. Δ_{rel} is a hyperparameter that determines the speed of increase of rejection rate. Defining λ_i as Equation 4 makes rejecting large loss samples in the loss function L. We do not reject any loss values at the first epoch, $t = 1$, since the model learns clean patterns in the initial phase. In practice, we use mini-batch in each iteration instead of full batch D' for composing the loss set. We call this method as LL-R.

Large loss correction (temporary). Another way of dealing with large loss sample is correcting rather than rejecting it. In a multi-label setting, this can be easily achieved by switching the corresponding annotation from negative to positive. Specifically, when the loss l_i is large and $i \in \mathcal{S}^u$, we *temporarily* modify its label to positive, i.e. $y_i^{AN} = 1$. The term "temporary" means that it does not change the actual label, but only uses the loss calculated from the modified one. To reflect this temporary correction scheme in Equation 3, we define the function λ_i as

$$\lambda_i = \begin{cases} \frac{\log f(\boldsymbol{x})_i}{\log(1 - f(\boldsymbol{x})_i)}, & i \in \mathcal{S}^u \text{ and } l_i > R(t) \\ 1, & \text{otherwise}\,, \end{cases} \tag{5}$$

where $R(t)$ is same as that in LL-R. This makes $l_i \times \lambda_i$ in Equation 3 to be the binary cross entropy loss between

the function output and positive label when $i \in \mathcal{S}^u$ and $l_i > R(t)$ because

$$\begin{aligned} l_i \times \lambda_i &= \text{BCELoss}(f(\boldsymbol{x})_i, y_i^{AN} = 0) \times \lambda_i \\ &= -\log(1 - f(\boldsymbol{x})_i) \times \lambda_i \\ &= -\log f(\boldsymbol{x})_i \\ &= \text{BCELoss}(f(\boldsymbol{x})_i, 1)\,. \end{aligned} \tag{6}$$

We name this method as LL-Ct. This method has the advantage that it increases the number of true positive labels from unobserved labels.

Large loss correction (permanent). In this method, we treat the large loss value more aggressively by *permanently* correcting the label. We directly change the label from negative to positive and use that modified label from the next training process. To achieve this, we define $\lambda_i = 1$ for every case, and modify the label as follows:

$$y_i^{AN} = \begin{cases} 1, & i \in \mathcal{S}^u \text{ and } l_i > R(t) \\ unchanged, & \text{otherwise}\,, \end{cases} \tag{7}$$

where $R(t)$ has a constant value of $\Delta_{rel}\%$ largest value in the loss set instead of $[(t-1) \cdot \Delta_{rel}]\%$. This makes the number of corrected labels gradually increase as the training progresses. When the label y_i^{AN} is modified by belonging to the first condition in Equation 7, the set $\mathcal{S}^u$ and $\mathcal{S}^p$ are also changed as follows:

$$\mathcal{S}^u \leftarrow \mathcal{S}^u - \{i\}\,, \tag{8}$$

$$\mathcal{S}^p \leftarrow \mathcal{S}^p \cup \{i\}\,. \tag{9}$$

We name this method as LL-Cp.

Absolute variant. Instead of gradually increasing the rejection/correction rate, we borrow the idea of using absolute value of loss as a rejection threshold [17] and apply it in WSML. In the rejection and temporary correction schemes, we define the function λ_i the same as Equation 4 except for $R(t)$ where it is defined as $R(t) = R_0 - t \cdot \Delta_{abs}$. R_0 and Δ_{abs} are hyperparameters where R_0 is an initial threshold and Δ_{abs} determines the speed of decrease of the threshold. We report the experimental results of these variant methods in Appendix.

4. Experiments

In this section, we present experimental results of our method and compare it with previous approaches in two different partial label setting in §4.1 and §4.2. In §4.3, we analyze the reason why our methods work well in 5 different ways, that is precision analysis, hyperparameter effect, qualitative results, model explanation, and generalization in a subset of training images. Throughout this section, we use mean average precision (mAP) as an evaluation metric.

4.1. Artificially created partial label dataset

Datasets. For a multi-label dataset where full labels are annotated, we artificially drop some of its labels for a partial label setting. Specifically, we follow the procedure presented by [7]: for each training image in a dataset, we randomly remain one positive label and regard other labels as unknown. We experiment on Pascal VOC 2012 [10], MS COCO 2014 [24], NUSWIDE [6], and CUB [42] datasets. For CUB the task is to classify not the bird categories but the attributes where multiple attributes exist for each image.
Implementation details. For fair comparisons we use the same seed number to create the same artificial dataset as in [7]. We use ResNet-50 [14] architecture which is pretrained on ImageNet [20] dataset. A single GPU with batch size 16 is used. Each image is resized into 448x448 and performed data augmentation by randomly flipping an image horizontally. We conduct experiments on two learning schemes. One is using the "LinearInit" which first freezes the backbone and update the weights of final linear layer for the initial epochs followed by fine-tuning the entire weights for the remaining epochs, and the other is "End-to-end" which is to fine-tune the entire weights from the beginning. Details about hyperparameter settings are described in Appendix.
Compared methods. We compare our method with Naive AN, Weak AN (WAN) [7, 27], Label Smoothing with AN (LSAN) [7, 37], EPR [7] and ROLE [7]. Note that some methods using only observed labels without using AN target (Curriculum labeling [9], IMCL [16]) doesn't work in this setting. They give a trivial solution that predicts all labels as positive since only positive labels are observed.
Results. As shown in Table 2, our method is closest to the fully labeled performance, e.g. 1.0 and 6.2 mAP difference in Pascal VOC and MS COCO datasets when fine-tuned end-to-end. Compared with Naive AN and Weak AN which use $\lambda_i = 0$ and $\lambda_i = \frac{1}{K-1}$ when $y_i^{AN} = 0$ in Equation 3, respectively, our three different methods all have better performance. Our method also surpasses LSAN in all datasets, especially having +4.1 and +2.7 mAP gain on a COCO dataset with End-to-end and LinearInit setting respectively. It implies that our method handles the label noise in AN target better than LSAN. Moreover, in most datasets, our method also outperforms EPR and ROLE. This result shows that gradually modifying large loss samples helps the model to have better generalization in the presence of false negative labels.

4.2. Real partial label dataset

Datasets. To see if our proposed method consistently works on a dataset with real partial label, we use OpenImages V3 [19] dataset where there is 3.4M training/42K validation/125K test images with 5,000 classes. In this dataset less than 1% of labels are annotated.

Method	End-to-end				LinearInit.			
	VOC	COCO	NUSWIDE	CUB	VOC	COCO	NUSWIDE	CUB
Full label	90.2	78.0	54.5	32.9	91.1	77.2	54.9	34.0
Naive AN	85.1	64.1	42.0	19.1	86.9	68.7	47.6	20.9
WAN [7, 27]	86.5	64.8	46.3	20.3	87.1	68.0	47.5	21.1
LSAN [7, 37]	86.7	66.9	44.9	17.9	86.5	69.2	50.5	16.6
EPR [7]	85.5	63.3	46.0	20.0	84.9	66.8	48.1	21.2
ROLE [7]	87.9	66.3	43.1	15.0	88.2	69.0	**51.0**	16.8
LL-R (Ours)	**89.2**	**71.0**	47.4	19.5	**89.4**	**71.9**	49.1	21.5
LL-Ct (Ours)	89.0	70.5	48.0	**20.4**	89.3	71.6	49.6	**21.8**
LL-Cp (Ours)	88.4	70.7	**48.3**	20.1	88.3	71.0	49.4	21.4

Table 2. **Quantitative results in artificially created partial label datasets.** Results of the model trained with full label are given in the second row to show the upper bound of WSML. "End-to-end" indicates that the entire weights of the model is fine-tuned from the beginning, while "LinearInit." indicates the backbone is frozen for the first few epochs. LL-Ct outperforms all baseline methods in 7 out of 8 settings, while LL-R and LL-Cp in 6 out of 8 settings.

Implementation details. We use ImageNet-pretrained ResNet-101 architecture and 4 GPUs with batch size 288. Each image is resized into 224x224 and random horizontal flip is applied during training. To better analyze the results, we sort the 5000 categories in ascending order with respect to the number of counted training images and divide them into 5 groups, having 1000 categories for each. Group1 is the group where the number of counted images are the smallest while Group5 is the biggest. We report the mAP results in each group as well as in all groups. Details about hyperparameter settings are described in Appendix.

Compared methods. We compare our method with Curriculum labeling [9], and IMCL [16], Naive AN, WAN and LSAN. Naive IU (Ignore Unobserved) is also compared which trains the model only with partial label. Note that ROLE [7] do not work because they require storing whole label matrix in a memory which is infeasible.

Results. The results are reported in Table 3. We first observe that training the model with naive BCE loss with AN target (Naive AN) boosts the classification performance for a large margin compared to previous methods using only the observed labels (Naive IU, Curriculum, IMCL). We speculate this performance improvement occurred since the average number of observed categories for each image is much smaller than the number of full categories, which hinders the model to be generalized to unseen data when trained with a limited amount of observed labels only. In contrast, even though the AN target is noisy, a large amount of categories may be annotated as true negative after modifying the unobserved labels to the negative labels, making the generalization performance of Naive AN better.

We also observe that LL-Ct has the best performance of 82.6 mAP, and other methods of ours provide similar high performance. Compared to the Naive AN, our method further rejects or corrects the possible false negative labels, making the degree of noisy labels as less as possible which leads to performance improvement in every groups, from Group1 to Group5. One thing to note is that WAN and LSAN show worse performance than Naive AN, which means that they cannot handle the label noise in AN target in a real partial label scenario.

Method	G1	G2	G3	G4	G5	All Gs
Naive IU	69.5	70.3	74.8	79.2	85.5	75.9
Curriculum [9]	70.4	71.3	76.2	80.5	86.8	77.1
IMCL [16]	71.0	72.6	77.6	81.8	87.3	78.1
Naive AN	77.1	78.7	81.5	84.1	88.8	82.0
WAN [7, 27]	71.8	72.8	76.3	79.7	84.7	77.0
LSAN [7, 37]	68.4	69.3	73.7	77.9	85.6	75.0
LL-R (Ours)	77.4	79.1	82.0	84.5	89.5	82.5
LL-Ct (Ours)	77.7	79.3	82.1	84.7	89.4	**82.6**
LL-Cp (Ours)	77.6	79.1	81.9	84.6	89.4	82.5

Table 3. **Quantitative results in OpenImages V3 dataset with real partial label.** 5000 categories are sorted in ascending order with respect to the number of training images in which the label of that category is known and then sequentially grouped from Group1 to Group5 with all groups having the same size. All Gs corresponds to the set of all categories. We observe that LL-Ct has the best performance, followed by LL-Cp and LL-R.

4.3. Analysis

In this section, we analyze the reason why our method works well in WSML. Unless mentioned, we perform analysis of our method on an artificially created COCO partial label dataset presented in §4.1 with $\Delta_{rel} = 0.2$.

Precision analysis. To verify whether the label that our proposed methods reject (LL-R) or correct (LL-Ct, LL-Cp) is

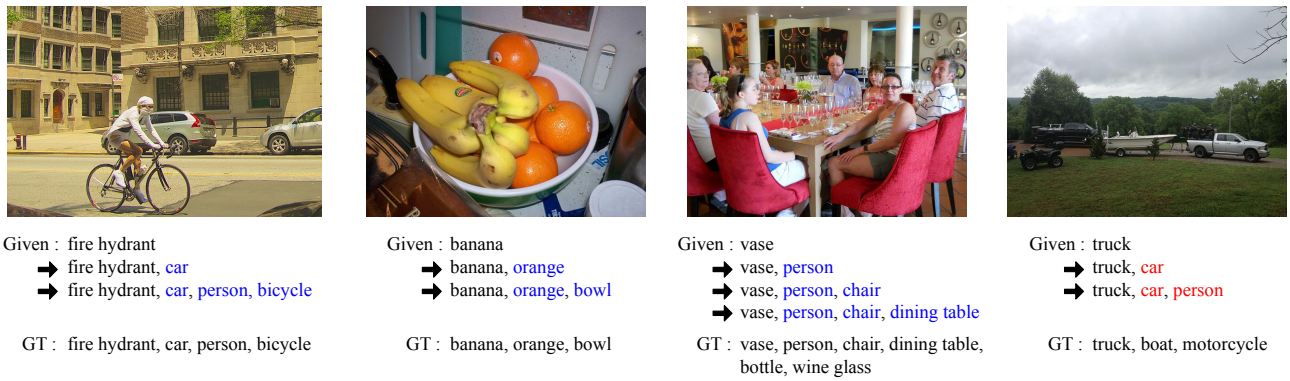

Figure 3. **Qualitative results in artificially generated COCO partial label dataset.** The arrow indicates the change of categories with positive label during training in our correction scheme LL-Ct and GT indicates actual ground truth positive labels for a training image. We show three cases where LL-Ct modifies the unannotated ground truth label correctly, and the failure case at the fourth column.

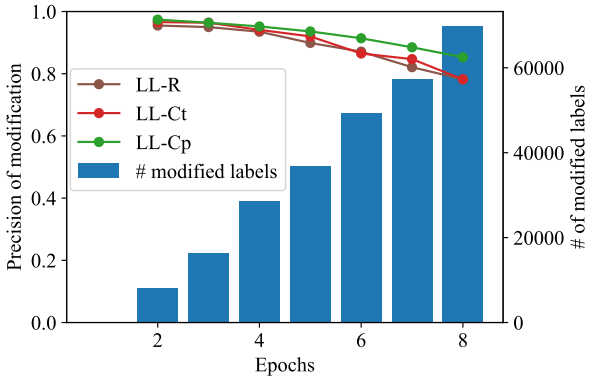

Figure 4. **Precision analysis of proposed methods on COCO dataset.**

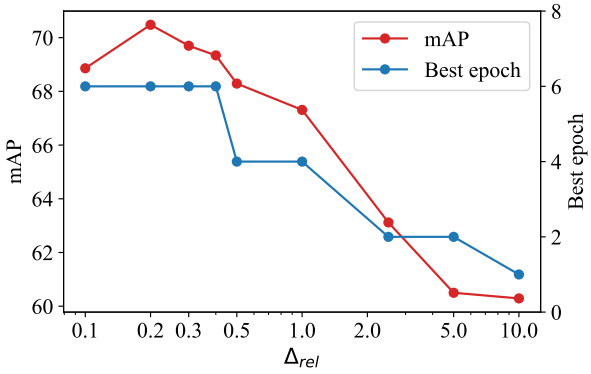

Figure 5. **Hyperparameter effect of LL-Ct on COCO dataset.**

actually noisy, we measure the precision of modification. That is, among the labels modified by our scheme as its loss values are large, we calculate the percentage of labels whose actual label is positive. While the precision is calculated in each epoch for LL-R and LL-Ct, we calculate the precision using the accumulated number of labels for LL-Cp for a fair comparison. We observe in Figure 4 that our schemes indeed modify the false negative labels with high precision. As the number of epoch increases, precision decreases because the model gradually memorizes the wrong label.

We can see that LL-Cp shows the highest precision value among our proposed schemes. However, according to Table 2, LL-Cp does not always guarantee highest performance and it may seem a bit contradictory. We conjecture that this is because of the characteristics of LL-Cp. As LL-Cp performs permanent correction, erroneously corrected labels may keep damaging the model learning once it is changed. Therefore, it might lead to lower mAP even with higher precision of modification.

Effect of hyperparameter Δ_{rel}. We evaluate the model performance of LL-Ct with different values of hyperparameter Δ_{rel} on a COCO dataset. From Figure 5 we observe that the model produces the best mAP when $\Delta_{rel} = 0.2$. When Δ_{rel} becomes smaller, its performance decreases because the model memorizes false negative labels that are not corrected due to a low correction rate. On the other hand, the performance decreases as Δ_{rel} increases after 0.2. Also, the number of epoch when the model has the best validation score decreases at this time. This is because as Δ_{rel} increases, our correction scheme wrongly modifies the true negatives labels as positive, making them false positives. The increased number of false positives hinders the model's generalization, letting model perform early stopping.

Qualitative results. Fig 3 shows the qualitative result of LL-Ct. The arrow indicates the change of categories with positive labels during training and GT indicates actual ground truth positive labels for a training image. We see that although not all ground truth positive labels are given, our proposed method progressively corrects the category of unannotated GT as positive. We also observe in the first three columns that a category that has been corrected once continues to be corrected in subsequent epochs, even though we perform correction temporarily for each epoch.

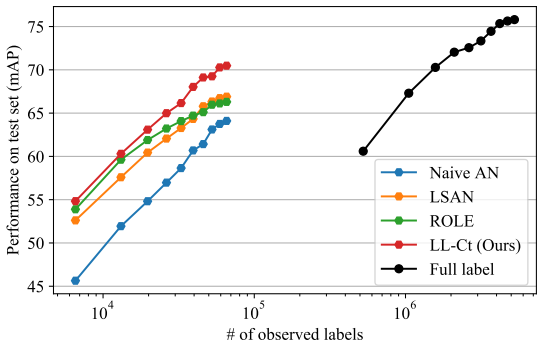

Figure 6. **Training with smaller number of image.**

This conveys that LL-Ct successfully keeps the model from memorizing false negatives. We also report the failure case of our method on the rightmost side where the model confuses the car as truck which is a similar category and misunderstands the absent category person as present.

Model explanation. We have seen that our methods have quantitatively better performance than other baseline methods. To see if this is related to the model's better understanding of the data, we examine how much the model's explanation is related to the human reasoning process. Concisely, we regard the class activation mapping (CAM) [58] as the model's explanation and the ground truth object as the human's explanation. To measure how much these two explanations are aligned, we use the Pointing Game metric [30, 57]. For each existing category in an input instance, we consider it as 'Hit' if the pixel point of the maximum value in CAM is inside the bounding box of the object, and 'Miss' if it is not. We count the #Hit and #Miss in all existing categories in all test data, and report the average of #Hit / (#Hit + #Miss) $\times$ 100 calculated for each category in Table 4.

Method	VOC	COCO
Naive AN	78.9	46.4
WAN [7, 27]	79.8	47.7
LSAN [7, 37]	79.5	49.1
EPR [7]	80.2	48.1
ROLE [7]	82.5	51.5
LL-R (Ours)	**83.7**	54.0
LL-Ct (Ours)	**83.7**	**54.1**
LL-Cp (Ours)	83.5	53.3

Table 4. **Pointing Game.**

We observe that in both VOC and COCO datasets, our three methods outperform previous methods. In particular, LL-Ct has +1.2 and +2.6 gain in VOC and COCO datasets compared to ROLE [7], respectively. This result indicates that the explanation of the model trained with our methods is better aligned to human's explanation. We report the CAM visualization results in Appendix.

Training with smaller number of image. To see if our method also works in a smaller number of training image, we randomly subsample training images in COCO dataset by 10%, 20%, $\cdots$, 90%, 100%, respectively, and train the model with partial label in §4.1 as well as full label. We then measure the classification performance on test set.

The results are shown in Figure 6. While the number of observed labels for weakly supervised methods with 100% of training image is much more smaller than the fully supervised method with 10% of training image, i.e. $\times$ 1/8, all the weakly supervised methods outperform the performance with full supervision. Moreover, LL-Ct showes a similiar performance to the fully supervised method with 30% training image only with 1/24 of the observed labels. This indicates that when we have a limited cost to annotate the labels when making a multi-label dataset, it is better to weakly annotate many images rather than fully annotate small number of images. We also observe that LL-Ct outperforms other weakly supervised methods on all ranges of number of observed labels. When only 10% of training image is given, LL-Ct has +9.2 mAP better performance compared to the result from Naive AN method. This means our method also provides better generalization with small number of training image.

5. Conclusion

In this paper, we present large loss modification schemes that reject or correct the large loss samples appearing during training the multi-label classification model with partially labeled annotation. This originates from our empirical observation that memorization effect also happens in a noisy multi-label classification scenario. Although heavy and complex components are not included, our schemes successfully keep the multi-label classification model from memorizing the noisy false negative labels, achieving state-of-the-art performance on various partially labeled multi-label datasets.

Limitations and broader impact. Since it is difficult to collect enormous data with fully annotated categories, partial label setting is essential [28, 35]. For instance, Instagram dataset is composed of billions of social media images with its corresponding hashtags as labels that are used to be noisy [28]. Our methodology makes one step progress towards dealing with noisy multi-label classification. However, current WSML methods have limitations that are yet to be reached to the performance with fully annotated label. We hope our methodology facilitates further research in the field of WSML to reach full label performance.

Acknowledgements. This work is in part supported by National Research Foundation of Korea (NRF, 2021R1A4A1030898(10%)), Institute of Information & communications Technology Planning & Evaluation (IITP, 2021-0-00106 (50%), 2021-0-01059 (20%), 2021-0-00180 (20%)) grant funded by the Ministry of Science and ICT (MSIT), Tech Incubator Program for Startups Korea, Ministry of SMEs and Startups, INMAC, and BK21-plus. Also, this work has been partially funded by the ERC (853489 - DEXIM) and by the DFG (2064/1 – Project number 390727645).

References

[1] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *ICML*, pages 233–242. PMLR, 2017. 2, 3

[2] Serhat Selcuk Bucak, Rong Jin, and Anil K Jain. Multi-label learning with incomplete class assignments. In *CVPR*, pages 2801–2808. IEEE, 2011. 1, 2

[3] Minmin Chen, Alice Zheng, and Kilian Weinberger. Fast image tagging. In *ICML*, pages 1274–1282. PMLR, 2013. 2

[4] Tianshui Chen, Muxin Xu, Xiaolu Hui, Hefeng Wu, and Liang Lin. Learning semantic-specific graph representation for multi-label image recognition. In *ICCV*, pages 522–531, 2019. 2

[5] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *CVPR*, pages 5177–5186, 2019. 2

[6] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. Nus-wide: a real-world web image database from national university of singapore. In *Proceedings of the ACM international conference on image and video retrieval*, pages 1–9, 2009. 2, 5

[7] Elijah Cole, Oisin Mac Aodha, Titouan Lorieul, Pietro Perona, Dan Morris, and Nebojsa Jojic. Multi-label learning from single positive labels. In *CVPR*, pages 933–942, 2021. 1, 2, 5, 6, 8

[8] Jia Deng, Olga Russakovsky, Jonathan Krause, Michael S Bernstein, Alex Berg, and Li Fei-Fei. Scalable multi-label annotation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3099–3102, 2014. 2

[9] Thibaut Durand, Nazanin Mehrasa, and Greg Mori. Learning a deep convnet for multi-label classification with partial labels. In *CVPR*, pages 647–657, 2019. 1, 2, 5, 6

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010. 2, 3, 5

[11] Hao Guo and Song Wang. Long-tailed multi-label visual recognition by collaborative training on uniform and rebalanced samplings. In *CVPR*, pages 15089–15098, 2021. 2

[12] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, pages 5356–5364, 2019. 1

[13] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W Tsang, and Masashi Sugiyama. Co-teaching: robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, pages 8536–8546, 2018. 2, 3, 4

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3, 5

[15] Jun Huang, Guorong Li, Qingming Huang, and Xindong Wu. Learning label specific features for multi-label classification. In *2015 IEEE International Conference on Data Mining*, pages 181–190. IEEE, 2015. 2

[16] Dat Huynh and Ehsan Elhamifar. Interactive multi-label cnn learning with partial labels. In *CVPR*, pages 9423–9432, 2020. 1, 2, 5, 6

[17] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, pages 2304–2313. PMLR, 2018. 2, 3, 5

[18] Ashish Kapoor, Raajay Viswanathan, and Prateek Jain. Multilabel classification using bayesian compressed sensing. *NeurIPS*, 25:2645–2653, 2012. 1, 2

[19] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*, 2017. 1, 2, 5

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 25:1097–1105, 2012. 5

[21] Kaustav Kundu and Joseph Tighe. Exploiting weakly supervised visual patterns to learn from partial annotations. *NeurIPS*, 33:561–572, 2020. 1, 2

[22] Jack Lanchantin, Tianlu Wang, Vicente Ordonez, and Yanjun Qi. General multi-label image classification with transformers. In *CVPR*, pages 16478–16488, 2021. 2

[23] Jisoo Lee and Sae-Young Chung. Robust training with ensemble consensus. In *ICLR*, 2019. 2, 3

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 2, 3, 5

[25] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *NeurIPS*, 33, 2020. 2

[26] Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? In *ICML*, pages 6448–6458. PMLR, 2020. 2

[27] Oisin Mac Aodha, Elijah Cole, and Pietro Perona. Presence-only geographical priors for fine-grained image classification. In *ICCV*, pages 9596–9606, 2019. 5, 6, 8

[28] Dhruv Kumar Mahajan, Ross B. Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018. 1, 8

[29] Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. Self: Learning to filter noisy labels with self-ensembling. In *ICLR*, 2019. 2

[30] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. In *BMVC*, 2018. 8

[31] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine learning*, 85(3):333–359, 2011. 2

[32] Tal Ridnik, Emanuel Ben-Baruch, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. Asymmetric loss for multi-label classification. In *ICCV*, pages 82–91, 2021. 1, 2

[33] Vaishaal Shankar, Rebecca Roelofs, Horia Mania, Alex Fang, Benjamin Recht, and Ludwig Schmidt. Evaluating machine accuracy on imagenet. In *ICML*, pages 8634–8644. PMLR, 2020. 1

[34] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. Selfie: Refurbishing unclean samples for robust deep learning. In *ICML*, pages 5907–5915. PMLR, 2019. 2

[35] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, pages 843–852, 2017. 1, 8

[36] Yu-Yin Sun, Yin Zhang, and Zhi-Hua Zhou. Multi-label learning with weak label. In *AAAI*, 2010. 1, 2

[37] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016. 5, 6, 8

[38] Farbound Tai and Hsuan-Tien Lin. Multilabel classification with principal label space transformation. *Neural Computation*, 24(9):2508–2542, 2012. 2

[39] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *CVPR*, pages 5552–5560, 2018. 2

[40] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *CVPR*, pages 9446–9454, 2018. 3

[41] Deepak Vasisht, Andreas Damianou, Manik Varma, and Ashish Kapoor. Active learning for sparse bayesian multilabel classification. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 472–481, 2014. 2

[42] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 2, 5

[43] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. Cnn-rnn: A unified framework for multi-label image classification. In *CVPR*, pages 2285–2294, 2016. 2

[44] Qifan Wang, Bin Shen, Shumiao Wang, Liang Li, and Luo Si. Binary codes embedding for fast image tagging with incomplete labels. In *ECCV*, pages 425–439. Springer, 2014. 2

[45] Xinshao Wang, Yang Hua, Elyor Kodirov, David A Clifton, and Neil M Robertson. Proselflc: Progressive self label correction for training robust deep neural networks. In *CVPR*, pages 752–761, 2021. 2

[46] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *CVPR*, pages 13726–13735, 2020. 2

[47] Baoyuan Wu, Siwei Lyu, and Bernard Ghanem. Ml-mg: Multi-label learning with missing labels using a mixed graph. In *ICCV*, pages 4157–4165, 2015. 2

[48] Tong Wu, Qingqiu Huang, Ziwei Liu, Yu Wang, and Dahua Lin. Distribution-balanced loss for multi-label classification in long-tailed datasets. In *ECCV*, pages 162–178. Springer, 2020. 2

[49] Miao Xu, Rong Jin, and Zhi-Hua Zhou. Speedup matrix completion with side information: Application to multi-label learning. In *NeurIPS*, pages 2301–2309, 2013. 2

[50] Hao Yang, Joey Tianyi Zhou, and Jianfei Cai. Improving multi-label learning with missing labels by structured semantic correlations. In *ECCV*, pages 835–851. Springer, 2016. 1, 2

[51] Yazhou Yao, Zeren Sun, Chuanyi Zhang, Fumin Shen, Qi Wu, Jian Zhang, and Zhenmin Tang. Jo-src: A contrastive approach for combating noisy labels. In *CVPR*, pages 5192–5201, 2021. 2

[52] Vacit Oguz Yazici, Abel Gonzalez-Garcia, Arnau Ramisa, Bartlomiej Twardowski, and Joost van de Weijer. Orderless recurrent models for multi-label classification. In *CVPR*, pages 13440–13449, 2020. 2

[53] Jin Ye, Junjun He, Xiaojiang Peng, Wenhao Wu, and Yu Qiao. Attention-driven dynamic graph convolutional network for multi-label image recognition. In *ECCV*, pages 649–665. Springer, 2020. 2

[54] Kun Yi and Jianxin Wu. Probabilistic end-to-end noise correction for learning with noisy labels. In *CVPR*, pages 7017–7025, 2019. 2

[55] Ying Yu, Witold Pedrycz, and Duoqian Miao. Multi-label classification by exploiting label correlations. *Expert Systems with Applications*, 41(6):2989–3004, 2014. 2

[56] Sangdoo Yun, Seong Joon Oh, Byeongho Heo, Dongyoon Han, Junsuk Choe, and Sanghyuk Chun. Re-labeling imagenet: from single to multi-labels, from global to localized labels. In *CVPR*, pages 2340–2350, 2021. 1

[57] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10):1084–1102, 2018. 8

[58] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929, 2016. 8