Interactive Multi-Label CNN Learning with Partial Labels

Dat Huynh
Northeastern University
huynh.dat@husky.neu.edu

Ehsan Elhamifar
Northeastern University
eelhami@ccs.neu.edu

Abstract

We address the problem of efficient end-to-end learning a multi-label Convolutional Neural Network (CNN) on training images with partial labels. Training a CNN with partial labels, hence a small number of images for every label, using the standard cross-entropy loss is prone to overfitting and performance drop. We introduce a new loss function that regularizes the cross-entropy loss with a cost function that measures the smoothness of labels and features of images on the data manifold. Given that optimizing the new loss function over the CNN parameters requires learning similarities among labels and images, which itself depends on knowing the parameters of the CNN, we develop an efficient interactive learning framework in which the two steps of similarity learning and CNN training interact and improve the performance of each another. Our method learns the CNN parameters without requiring keeping all training data in the memory, allows to learn few informative similarities only for images in each mini-batch and handles changing feature representations. By extensive experiments on Open Images, CUB and MS-COCO datasets, we demonstrate the effectiveness of our method. In particular, on the large-scale Open Images dataset, we improve the state of the art by 1.02% in mAP score over 5,000 classes.

1. Introduction

Finding all labels in an image, referred to as multi-label recognition [1, 2, 3], is a fundamental learning problem with a wide range of applications, including self-driving cars, surveillance systems and assistive robots. While deep Convolutional Neural Networks (CNNs) have shown great performance for single-label image classification, their adaption to the multi-label recognition faces major challenges, especially in real problems with a large number of labels.

First, training multi-label CNNs requires collecting multi-label annotations for a large number of images, which is significantly more difficult than single-label annotations [4]. In fact, many existing multi-label datasets, such as MS-COCO [5], YahooFlickr [6] and Open Images [7], contain only small partial labels of images. As a result, multi-label learning methods that assume access to full labels of images [8, 9, 10] are not applicable. Moreover, training CNNs by treating missing labels as negatives [2, 11, 12, 13, 14, 15] results in significant performance drop as many ground-truth positive labels are falsely labeled [16, 17]. On the other hand, adapting CNNs to the multi-label classification by simply transforming it into multiple single-label classification problems and training via the ranking [18] or cross-entropy [19] loss fails to model the dependencies among labels, which is particularly important for handling partial labels. Finally, multi-label learning methods that handle partial labels using low-rank learning [20, 21, 22, 23] or semi-supervised learning [24, 25] generally do not allow end-to-end training as they require knowing and fixing the feature representation of images to learn classifier parameters, or require solving a costly optimization problem with all training data in memory.

In this paper, we develop an efficient framework for end-to-end training of multi-label CNNs with partial labels by learning and leveraging dependencies among labels and images in an interactive scheme. We introduce a new loss function that regularizes the standard binary cross-entropy loss with a cost function that measures the smoothness of labels and features of images on the data manifold. Given that optimizing the new loss function over the CNN parameters requires learning similarities among labels and images, which itself depends on knowing the parameters of the CNN, we develop an efficient interactive learning scheme in which the two steps of similarity learning and CNN training interact and improve the performance of each another, see Figure 1. More specifically, fixing the CNN, we learn label and image dependencies by minimizing the smoothness loss. Fixing dependencies, we optimize the total loss over CNN parameters and repeat the two steps until convergence.

Our method allows to learn the CNN in an end-to-end fashion without requiring keeping all training data in the memory. Unlike expensive graph-based learning algorithms that require building and operating on the entire graph adjacency or laplacian [22, 25, 23], our method allows to learn few informative similarities only for images in each

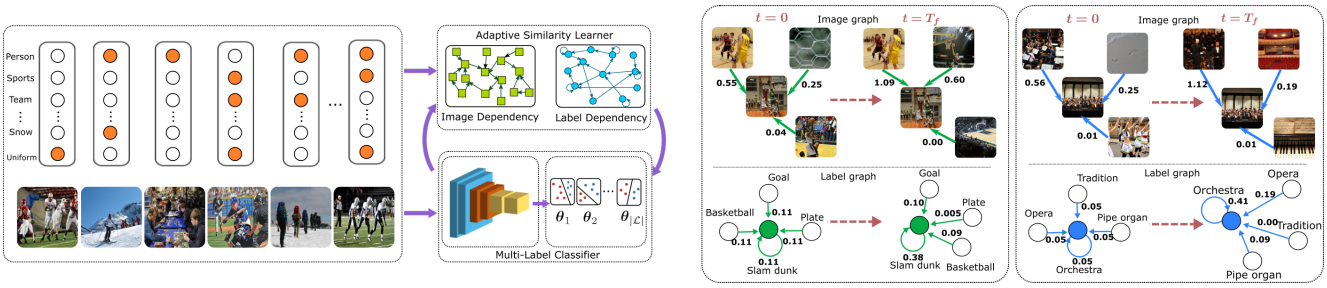

Figure 1: Left: Our proposed semi-supervised multilabel recognition framework consists of a CNN classifier and an adaptive similarity learner that interact and improve the performance of each other during training. Right: Visualization of the learned image and label similarity graphs via interaction with the CNN during training on the Open Images dataset. We show the image and label similarities learned by the initial CNN ($t = 0$) and the final similarities learned at the last interactive learning step ($t = T_f$).

mini-batch and to handle changing feature representations. Our method borrows ideas from semi-supervised learning, however, unlike semi-supervised multi-label learning, it allows to update feature representation of images and handles training data with partial labels. By extensive experiments, we show that our framework outperforms the state of the art, in particular, improving the mAP score on the large-scale Open Images dataset by 1.02% over 5,000 labels.

2. Related Work

The first line of work on multi-label learning treats each label prediction as an independent binary classification problem [26]. However, it is not scalable when the number of labels is large, treats missing labels as negatives which leads to performance drop and ignores dependencies among labels which is important for recognition. To overcome the last challenge, the majority of existing work on multi-label learning try to incorporate dependencies among labels. In particular, several methods use graphical models [8, 9, 10], by learning label occurrence and co-occurrence potential functions using Markov Random Fields. However, they require knowing the full labels of training data and have difficulty dealing with large number of labels as the number of parameters to learn will become prohibitively large. To deal with partial labels, several works treat missing labels as negative labels [2, 11, 12, 13, 14, 15, 27]. However, this could result in significant performance drop since many ground-truth positive labels are falsely annotated [16].

Semi-supervised multi-label learning, on the other hand, assumes access to a subset of images with full labels and a large number of images without labels or with partial noisy labels [28, 24]. When image and label dependencies are incorporated via label-label and image-image graphs [29], such methods require a known and fixed feature representation of data, which does not allow for feature learning or fine-tuning of CNNs. While [30] learns an adaptive graph for label propagation, it cannot generalize to novel images due to its transductive nature and cannot scale to large datasets. Moreover, the assumption of having a subset of images with full labels could be limiting, which is also different than the partial label setting considered in this paper, where all training images contain only a subset of ground-truth labels. Curriculum learning, self-training, also called bootstrapping [31, 32, 33, 34] tries to increase the number of labels by alternating between learning a binary classifier for each label using available annotations and adding unannotated images about whose label the classifier is most certain to training data. [3] further combines graph neural network and curriculum learning to capture label correlation while exploiting unlabeled data. However, curriculum learning, and self-training in general, suffer from semantic drift, since unannotated images that receive incorrect labels are permanently added to training data. To mitigate this issue, constrained bootstrapping [32] incorporates positive and negative dependencies among labels. However, it requires building complete graphs among images and attribute classifiers, which are hard to obtain and train when dealing with a large number of labels and images.

To effectively handle partial labels, [22] encodes a network of label dependencies via a mixed graph, while [4, 14] learns correlation between labels to predict some missing labels. On the other hand, [23] generalizes the linear correlation assumption to structured semantic correlations. Several methods treat missing labels as hidden variables via probabilistic models and predict missing labels by posterior inference [35, 36, 37]. The work in [38] models missing labels as negatives and corrects the induced error by learning a transformation on the output of the multi-label classifier that models the labeling bias. Orthogonal to these directions, [39, 40, 41] exploit correlations among labels and among images with sparse/low-rank regularization to complete the image-label matrix, while [20] formulates the problem as a low-rank empirical risk minimization. However, the majority of these work cannot be used to learn a deep CNN as they require knowing and fixing the features of images, require keeping all training data in memory, or require solving a costly optimization which is not scalable to large datasets. In this paper, we develop a framework that allows efficient end-to-end CNN training with partial labels and is scalable to large number of labels and images.

Remark 1 *Notice that the work on partial multi-label learning in [42, 43], which assume that all missing labels are negative and a subset of positive labels are true, is different than the partial label setting studied in this paper, where the missing labels could be positive or negative.*

3. Interactive Multi-Label CNN Learning

We consider the multi-label recognition problem via CNN, whose goal is to find all labels of an image. Assume we have N training images $I_1, \ldots, I_N$, for each we observe a few positive and negative labels with the values of many labels missing. Let $\mathcal{C}$ be the set of all labels. For an image i, we denote the set of its observed labels by $\Omega_i \subseteq \mathcal{C}$ and the values of observed labels by $\{y^o_{j,i}\}_{j\in\Omega_i}$, where $y^o_{j,i} \in \{-1, +1\}$ indicates the presence $(+1)$ or absence (-1) of the label j in the image i. Our goal is to find the complete label vector $\boldsymbol{y}_i \in \{-1, +1\}^{|\mathcal{C}|}$ of each image i and effectively train a multilabel CNN, given the small number of positive and negative images for every label.

Let $\boldsymbol{w}$ denote the parameters of the CNN up to the feature extraction layer (layer before the last) and $\{\boldsymbol{\theta}_j\}_{j=1}^{|\mathcal{C}|}$ denote parameters of the $|\mathcal{C}|$ logistic regression models in the last layer of the CNN. We denote by $\boldsymbol{f}^i_{\boldsymbol{w}} \triangleq f_{\boldsymbol{w}}(I_i)$ the feature vector of image i.

3.1. Proposed Framework

We propose an efficient framework for multi-label CNN learning with partial labels that consists of two components: a multi-label CNN classifier and an adaptive similarity learner. The similarity learning discovers the dependencies among labels and among images using the current knowledge of CNN. We use the learned similarities to define a prediction smoothness loss that regularizes training the CNN via the standard binary cross-entropy loss using available labels. More specifically, to learn the parameters of the network, $\big(\boldsymbol{w}, \{\boldsymbol{\theta}_j\}_{j=1}^{|\mathcal{C}|}\big)$, we propose to minimize the following loss function

$$\min_{\boldsymbol{w},\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_{|\mathcal{C}|}} \sum_i \mathcal{L}_{\mathrm{c}}^{(i)}\Big(\boldsymbol{w}, \{\boldsymbol{\theta}_j\}_{j=1}^{|\mathcal{C}|}\Big) + \mathcal{L}_{\mathrm{s}}^{(i)}\Big(\boldsymbol{w}, \{\boldsymbol{\theta}_j\}_{j=1}^{|\mathcal{C}|}\Big), \tag{1}$$

where $\mathcal{L}_c^{(i)}$ is the cross-entropy classification loss for image i, which is defined by observed image labels $\{y^o_{j,i};\ i = 1, \ldots, N,\ j \in \Omega_i\}$ as

$$\mathcal{L}_{\mathrm{c}}^{(i)} \triangleq -\sum_{j\in\Omega_i} y^o_{j,i} \log(p_{j,i}) + (1 - y^o_{j,i}) \log(1 - p_{j,i}), \tag{2}$$

where $p_{j,i}$ is the output of the classifier j for the image i. On the other hand, $\mathcal{L}_s^{(i)}$ is a smoothness loss that enforces the predicted labels and learned feature of image i to be smooth on the data manifold according to learnable label and image similarities, which we discuss next.

3.1.1 Label and Image Dependency Smoothness Loss

Given partial labels and the small amount of annotations for each label, training the multi-label CNN is prone to overfitting. Thus, we regularize training by considering a loss function $\mathcal{L}_{\mathrm{s}}(\cdot)$ that constrains predictions to be smooth according to dependencies and similarities among labels and among images.

Label Dependency Smoothness. We start by using a label graph whose structure is known, capturing dependencies among labels, yet its connections weights will be learned. To do so, we measure the co-occurrence rate of each pair of labels in the training set. For each label, we choose the k_a most co-occurred labels to connect to. We exploit the structure of the graph to constrain training pairs of classifiers that are connected by edges, while learning the connection weights through our framework. To be specific, let

$$y_{j,i} = y^o_{j,i},\ j \in \Omega_i,\ \ y_{j,i} = 2p_{j,i} - 1,\ j \notin \Omega_i, \tag{3}$$

where we convert $p_{j,i} \in [0, 1]$, which is the probability of image i having label j, to $y_{j,i} \in [-1, +1]$. In other words, we impute the missing labels using CNN. The label-label matrix $\boldsymbol{A} \triangleq [a_{j',j}] \in \mathbb{R}^{|\mathcal{C}|\times|\mathcal{C}|}$, whose non-zero support is known but its weights $\{a_{j',j}\}_{j,j'\in\mathcal{C}}$ are unknown, denotes dependency strengths. Let $\mathcal{Q}_j$ denote the set of neighbors of the label j on the graph, i.e., the set of labels related to label j. We propose a model in which each label of an image can be determined by the related labels of semantically similar images. More specifically, we consider the model

$$y_{j,i} = \tanh\Big(\sum_{i'} c_{i',i} \sum_{j'\in\mathcal{Q}_j} a_{j',j} y_{j',i'}\Big), \tag{4}$$

in which the label j of image i is determined by a linear combination of neighboring labels j' (with coefficients $a_{j',j}$) over semantically related images i' (with coefficients $c_{i',i}$). Here $c_{i',i}$ denotes the degree of the semantic similarity of i' to i. The tangent hyperbolic function, $\tanh$, maps the result to $[-1, +1]$. As an example, if an image i' containing the label 'slam dunk' is similar to an image i, we expect 'slam dunk' and its related labels such as 'basketball' to also apear in i. Thus, we define the label smoothness loss as

$$\ell_y^{(i)} \triangleq \Big\|\boldsymbol{y}_i - \tanh\Big(\sum_{i'} c_{i',i} \boldsymbol{A}\boldsymbol{y}_{i'}\Big)\Big\|_2^2, \tag{5}$$

to measure the error associated with (4), which is rewritten in the vector form.

Image Dependency Smoothness. Complement to the label smoothness loss, we also define a feature smoothness loss to enforce smoothness on the image manifold. We assume that similar images, which contain many shared/similar labels, have similar visual features. More specifically, we model

that the feature vector of each image can also approximately be written as a linear combination of feature vectors of semantically similar images, and define

$$\ell_f^{(i)} \triangleq \Big\| \boldsymbol{f}_i - \sum_{i'} \bar{c}_{i',i} \boldsymbol{f}_{i'} \Big\|_2^2. \tag{6}$$

The coefficients $\{\bar{c}_{i',i}\}$ denote the similarities between image features. While the similarity coefficients $c_{i',i}$ and $\bar{c}_{i',i}$ take different values, they both must give rise to selecting the same images as being semantically similar to an image i, i.e., they must have the same nonzero support.

We define the smoothness loss function by combining the losses in (5) and (6),

$$\mathcal{L}_s^{(i)}\Big(\boldsymbol{w}, \{\boldsymbol{\theta}_j\}_{j=1}^{|\mathcal{C}|}\Big) \triangleq \min_{\substack{\{c_{i',i}, \bar{c}_{i',i}\} \in \mathcal{R} \\ \{a_{j',j}\}}} \lambda_y \ell_y^{(i)} + \lambda_f \ell_f^{(i)} \tag{7}$$

which requires optimizing over, hence learning, the image $\{c_{i',i}, \bar{c}_{i',i}\}$ and label $\{a_{j',j}\}$ similarities. Here, $\lambda_y, \lambda_f \geq 0$ are the regularization parameters (since we add the smoothness loss to the cross entropy loss in (1), we use two regularization parameters). The minimization must take into account that the similar images to each image i must be the same. Thus, we define the constraint set $\mathcal{R}$ as

$$\mathcal{R} \triangleq \Big\{ c_{j,i}, \bar{c}_{j,i} \geq 0, \sum_j \mathrm{I}\big(\big\|\big[c_{j,i},\ \bar{c}_{j,i}\big]\big\|\big) \leq k,\ \forall i, j \Big\}, \tag{8}$$

where $\mathrm{I}(\cdot)$ is an indicator function that is one when its argument is nonzero and is zero otherwise. Given that $c_{j,i}, \bar{c}_{j,i}$ are similarities, we enforce them to be nonnegative. The second constraint enforces that each image selects at most k other images as similar. Here, k is a hyperparameter.

Learning Similarities. To find label similarities, we perform gradient descent on the objective function of (7) with respect to $\{a_{j',j}\}$. To find image similarities, given the constraints in $\mathcal{R}$, we develop a novel framework by generalizing the Orthogonal Matching Pursuit (OMP) algorithm [44], proposed for sparse recovery of a single vector, to Joint Nonnegative OMP to find both $\{c_{i',i}\}$ and $\{\bar{c}_{i',i}\}$. Algorithm 2 shows the steps (see the supplementary materials for the derivations of the algorithm). For each point i, the algorithm starts by initializing an active set $\mathcal{S} = \varnothing$ and two residual vectors $\boldsymbol{r}_y = \boldsymbol{y}_i$ and $\boldsymbol{r}_f = \boldsymbol{f}_i$ (step 2), picking the point i' in the dataset that is best correlated to these two vector jointly (step 4) and adding it to $\mathcal{S}$. We then solve for the similarity values by minimizing ℓ_y and ℓ_f over the coefficients in $\mathcal{S}$ with thresholding them at zero (steps 9 and 10) and update the residuals accordingly (steps 11 and 12). Notice that we use a first-order approximation of the hyperbolic tangent function in $\mathcal{L}_s^{(i)}$, which is $\tanh(x) \approx x$, to efficiently solve for image and label similarities (see the supplementary material for more details).

Algorithm 1 : Interactive Multi-Label CNN Learning

Input: Training set $\big\{(I_i, \{y_{j,i}^o\}_{j \in \Omega_i})\big\}_{i=1,\ldots,N}$

1: Initialize CNN parameters $\boldsymbol{w}, \{\boldsymbol{\theta}_j\}_{j=1}^{|\mathcal{C}|}$
2: **repeat**
3: **Adaptive Similarity Learning:**
4: Fix parameters of CNN
5: Solve for similarities $\{c_{i',i}, \bar{c}_{i',i}\}$ in (7) via Algorithm 2
6: Solve for label weights $\{a_{j',j}\}$ in (7) via gradient descent
7: **Constrained CNN Learning:**
8: Fix image and label similarities
9: Train CNN via backpropagation on the loss function (1).
10: **until** convergence

Output: Optimal CNN parameters $(\boldsymbol{w}, \{\boldsymbol{\theta}_j\}_{j=1}^{|\mathcal{C}|})$, label and image similarities $\{\alpha_{j',j}\}, \{c_{i',i}, \bar{c}_{i',i}\}$

3.2. Interactive Learning Algorithm

Learning the parameters of CNN via minimization of (1) is not straightforward since computing each $\mathcal{L}_s^{(i)}$ requires solving for the label and image similarity coefficients that in turn requires knowing all labels $\{\boldsymbol{y}_i\}_{i=1}^N$ and features $\{\boldsymbol{f}_i\}_{i=1}^N$ of images, which are unknown.

To tackle the problem, we propose to minimize the loss function in (1) via an alternating optimization scheme, which leads to interactively learn the CNN parameters and improve the similarities over time, see Algorithm 1. More specifically, in the adaptive similarity learning step, given current CNN parameters, we compute the missing labels and solve (7) to find similarities. Given learned label and image similarities, in the constrained CNN training step, we train the parameters of the CNN via backpropagation on our new loss function in (1). We alternate between the two steps until either the cost function converges or the validation error does not decrease. Notice that we solve (1) over each minibatch via the interactive algorithm. Thus, the similarity graph is learned only for images in the current minibatch and we do not need to process the entire graph.

In our experiments, we initialize the classifier parameters $\{\boldsymbol{\theta}_j\}_{j=1}^{|\mathcal{C}|}$ by running logistic regression on available image annotations and initialize $\boldsymbol{w}$ using state-of-the-art convolutional networks, in our case, ResNet-101 [45] for Open Images and CUB experiments and VGG-16 [46] for MSCOCO experiments (see the experiments section for details).

Remark 2 *Our interactive learning framework allows some connections in the label graph to be removed, by setting their weights to zero, and some connections be less/more emphasized, by setting different weights to them during training. Also, it is worth noting that we do not necessarily require to have connections for every label; we could set the label graph to identity when labels are independent (as in the experiments on CUB).*

Algorithm 2 : Similarity Learning via Joint Nonnegative OMP

Input: $\{\boldsymbol{f}_i\}_{i=1}^N$, $\{\boldsymbol{y}_i\}_{i=1}^N$, label similarities $\{a_{j',j}\}$, number of nonzero entries k, regularization parameters λ_y, λ_f.

for $i = 1, \ldots, N$ **do**
  Initialize residuals $\boldsymbol{r}_y = \boldsymbol{y}_i, \boldsymbol{r}_f = \boldsymbol{f}_i$, similarity set $\mathcal{S} = \varnothing$
  for $t = 1, \ldots, k$ **do**
    $s = \text{argmax}_{i'} \lambda_y \frac{\langle \boldsymbol{r}_y, \boldsymbol{A}\boldsymbol{y}_{i'} \rangle^2}{\|\boldsymbol{A}\boldsymbol{y}_{i'}\|^2} + \lambda_f \frac{\langle \boldsymbol{r}_f, \boldsymbol{f}_{i'} \rangle^2}{\|\boldsymbol{f}_{i'}\|^2}$
    if $\langle \boldsymbol{r}_y, \boldsymbol{y}_s \rangle$ or $\langle \boldsymbol{r}_f, \boldsymbol{f}_s \rangle \leq 0$ **then**
      Break
    end if
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{s\}$
    $\{c_{i',i}\} = \max\big(0, \text{argmin} \|\boldsymbol{y}_i - \sum_{i' \in \mathcal{S}} c_{i',i} \boldsymbol{A}\boldsymbol{y}_{i'}\|_2^2\big)$
    $\{\bar{c}_{i',i}\} = \max\big(0, \text{argmin} \|\boldsymbol{f}_i - \sum_{i' \in \mathcal{S}} \bar{c}_{i',i} \boldsymbol{f}_{i'}\|_2^2\big)$
    $\boldsymbol{r}_y \leftarrow \boldsymbol{y}_i - \sum_{i' \in \mathcal{S}} c_{i',i} (\boldsymbol{A}\boldsymbol{y}_{i'})$
    $\boldsymbol{r}_f \leftarrow \boldsymbol{f}_i - \sum_{i' \in \mathcal{S}} \bar{c}_{i',i} \boldsymbol{f}_{i'}$
  end for
end for

Output: Similarities $\{c_{i',i}, \bar{c}_{i',i}\}$

Remark 3 *Unlike conventional graph-based semi-supervised methods that fix the graph and then regularize the training, in our framework, the two components interact and improve the performance of each other over time. Unlike curriculum labeling and self-training, our framework does not fix the label of selected unlabeled data, which can propagate prediction error, instead it regularizes the prediction to be globally consistent across training images.*

4. Experiments

We evaluate the performance of our proposed multi-label recognition framework on multiple datasets, including the large-scale Open Images [7], CUB-200-2011 [47] and MS-COCO [5] datasets.

4.1. Datasets

Open Images. The Open Images dataset (version 3) consists of 9 million training images as well as 41,620 and 125,436 images for validation and testing, respectively. The dataset has 5,000 trainable classes, where each class has at least 100 samples. Given the large number of images, classes and the fact that each image has only a few labels, we use this dataset to demonstrate the effectiveness of our framework on dealing with large datasets. We use the provided training, validation and testing splits in the dataset for training, hyperparameter tuning and testing of all methods.

CUB-200-2011. To systematically evaluate the performance of our method as a function of the fraction of missing labels, we use the CUB dataset, which is a fine-grained image dataset of 200 different bird species. Each image in the dataset has an 312-dimensional attribute vector, indicating the presence (1) or absence (-1) of an attribute in the image. We follow [48] for training, validation and testing split.

MS-COCO. We follow the experimental setup in [38], where we use approximately 80K images for training and 20k images for testing. 1000 most frequent words in captions are considered as training labels. For each image, we generate a 1000-dimensional vector indicating whether a label is present (+1) or absent (-1) in the image caption.

4.2. Baselines and Model Variants

We choose the **Logistic** regression model, which corresponds to minimizing our loss function in (1) with $\lambda_y = \lambda_f = 0$, using available labels in images. We fine-tune the networks end-to-end on available labels in images. *We use this baseline to initialize all methods in our experiments.* We compare with **Wsabie** [49], which models label correlation by measuring the inner product between class semantics and image features, as well as **CNN-RNN** [2], which uses Recurrent Neural Network to model high order label correlation and predicts next labels conditioned on all current present labels. We also compare with **Fast0Tag** [50], which learns a nonlinear transformation from image features to a semantic space. Following the recent advances in training CNNs with partial labels, we use **Curriculum Labeling**[1] with score thresholding strategy [3] as a strong baseline, which alternates between labeling unlabeled data with high prediction confidence and retraining classifiers on the updated training set. We include **Latent Noise** [38] that learns to correct the bias associated with missing labels by simultaneously training a relevance classifier, modeling the human labeling bias, and an unbiased visual classifier. Finally, we use **LSEP** [51], which uses a differentiable log-sum-exp pairwise loss, being easier to optimize than the traditional ranking loss for multi-label learning.

Our method. For our Interactive Multi-label CNN Learning (**IMCL**) method, we use the validation set of each dataset for tuning the hyperparameters, which are λ_y, λ_f in (7) and k in (8). This leads to setting $\lambda_y = 1, \lambda_f = 0.5, k = 5$ for the Open Images and $\lambda_c = 2, \lambda_f = 0.5, k = 5$ for CUB and $\lambda_y = \lambda_f = 0.5, k = 3$ for MS-COCO. For Open Images and MS-COCO, we set $k_a = 50$ to build the label graph, i.e, we connect each label to its top 50 co-occurred labels in the training set (results did not change by using similar values, as our method can set the weights to zero if needed), while for CUB, we set the label graph to the identity, given the independency of the attributes (labels).

4.3. Implementation Details

To have a fair comparison, for each dataset, we use the same CNN architecture as the feature extractor for all methods. On Open Images and CUB, we use ResNet-101 pre-trained on OpenImage and ImageNet, respectively. On MS-COCO, we follow [38] and use the pre-trained VGG-16 on

[1] We measure the performance on all 5000 labels which is different from [3] that only uses 600 labels.

Model	Group 1	Group 2	Group 3	Group 4	Group 5	All classes
Logistic	69.47	70.29	74.79	79.23	85.49	75.85
Latent Noise (relevance)	69.14 (69.25)	69.93 (69.75)	74.60 (74.57)	78.89 (78.85)	85.37 (85.29)	75.59 (75.54)
Latent Noise (visual)	69.37 (69.50)	70.41 (70.32)	74.79 (74.78)	79.20 (79.22)	85.51 (85.47)	75.86 (75.86)
CNN-RNN	68.76 (68.85)	69.70 (69.56)	74.18 (74.02)	78.52 (78.55)	84.61 (84.47)	75.16 (75.09)
LSEP	69.49 (69.49)	70.23 (70.23)	74.80 (74.81)	79.18 (79.19)	85.47 (85.47)	75.83 (75.84)
Fast0Tag	69.74 (69.58)	70.65 (70.41)	75.42 (75.01)	79.81 (79.41)	86.06 (85.73)	76.34 (76.03)
Wsabie	69.77 (69.23)	70.87 (70.10)	76.03 (75.06)	80.25 (79.42)	86.04 (85.50)	76.59 (75.86)
Curriculum Labeling	70.37 (69.77)	71.32 (70.86)	76.23 (75.45)	80.54 (79.62)	86.81 (85.91)	77.05 (76.32)
IMCL (Ours)	**70.95** (**69.91**)	**72.59** (**71.36**)	**77.64** (**75.94**)	**81.83** (**80.15**)	**87.34** (**86.32**)	**78.07** (**76.72**)

Table 1: mAP scores (%) of all methods with end-to-end training and fixed feature representation (in parenthesis) on the test set of the Open Images dataset.

ImageNet. We implement all methods in Tensorflow and optimize with RMSProp [52] with learning rate 0.001 on OpenImage and 0.01 on CUB and MS-COCO. We use exponential learning rate decay of 0.8 whenever the validation performance degrades. On MS-COCO, we reduce the learning rate to 0.001 after two epochs. We initialize all methods with the logistic model weights and refine them with $1, 3, 4$ epochs with batch size of $32, 32, 1$, respectively, on Open Images and CUB and MS-COCO. We also renormalize the value of y from the range of $[-1, +1]$ to $[-0.5, +1]$ so that similarity learning would focus more on positive labels instead of the majority negative labels in each image.

4.4. Evaluation metric

To evaluate the performance of different methods for multi-label learning, we measure the average precision (AP) for each class and mean AP over the dataset, similar to [28]. For each class, AP is computed as

$$AP_c = \frac{1}{N_c} \sum_{k=1}^{N} \text{Precision}(k, c) \cdot \text{rel}(k, c), \qquad (9)$$

where N_c is the number of images containing class c, $\text{Precision}(k, c)$ is the precision for class c when retrieving k best predictions and $\text{rel}(k, c)$ is the relevance indicator function that is 1 iff the class c is in the ground-truth of the image at rank k. We also compute the performance across all classes using mean average precision (mAP) defined as $mAP = 1/|\mathcal{C}| \sum_c AP_c$, where $|\mathcal{C}|$ is the number of classes.

4.5. Results on Open Images Dataset

We setup two experiments. In the first experiment, we fix the feature extractor $f_{\boldsymbol{w}}(\cdot)$ for all methods such that data representation does not change during training as in classical setting. In the second experiment, we train all models end-to-end. Through interactive learning, our model exploits the change in the data representation manifold, which significantly improves the performance as we show.

Effect of the Number of Training Images. To better analyze the effect of the number of available images for each label, we rank all classes in the ascending order with respect to the number of available annotations per class in the training set and divide them into 5 groups of equal size, where Group 1 corresponds to 1000 labels with the least number of available annotations and Group 5 corresponds to 1000 labels with the most number of annotations.

Table 1 shows the mAP scores of different methods on the test set of Open Images for each group and for all labels. The number before the parenthesis shows the mAP when training end-to-end and the number inside the parenthesis shows the performance when only classifier parameters are learned. As expected, the performance of all methods improves from Groups 1 to 5, since the number of training images for each label increases. While Logistic, LSEP and visual classifier of Latent Noise perform similarly on the entire dataset, as they only exploit labeled data, Wsabie and Fast0Tag slightly perform better as they exploit label correlation. Curriculum Labeling performs better than other baselines, as it takes advantage of unlabeled data for better recognition. On the other hand, our method without representation learning improves the mAP score on the dataset by 0.4%, thanks to both using unlabeled data and its ability to adaptively learn appropriate image and label similarity graphs for learning better visual models of different classes. When training all models end-to-end, our method obtains the most improvement compared with baselines, which indicates the effectiveness of our interactive learning. Notice that CNN-RNN, which treats missing labels as absent, obtains lower performance than other baselines. Overall, our method obtains 1.02% improvement with respect to the second best method, Curriculum Labeling.

Effect of Regularization Parameters. Table 2 shows the effect of the hyperparameters λ_y, λ_f, k on the mAP score for the *validation set*, which we use to select the best values. Notice that for a fixed λ_f (and similarly λ_c), the mAP score improves as we increase the regularization parameter and it decreases for large values of the regularization. In fact, the best score is obtained for $(\lambda_c = 1, \lambda_f = 0.5)$, demonstrating the effectiveness of both terms in (7) that use label and feature vectors for similarity learning. Also, the

Figure 2: Qualitative results for multilabel recognition by different algorithms on several images from the Open Images test set. A ground-truth label is considered as recognized if it is in the top 50 highest prediction for an images. We rank the labels according to how many methods are able to recognize them where the top label denotes the least recognizable among all methods. Our method manages to recognize small objects which are often ignored by others such as Arm or Mircophone in the first image and Surfboard in the second image. Our method also consistently improves label recalls across images by leveraging unlabeled data to better regularize prediction of rare labels.

λ_y	0	0.5	1	2
mAP ($\lambda_f = 0.5$)	78.12	78.37	**78.44**	78.40

λ_f	0	0.5	1	2
mAP ($\lambda_y = 1$)	78.40	**78.44**	78.38	78.17

k	3	5	7
mAP ($\lambda_y = 1, \lambda_f = 0.5$)	78.34	**78.44**	78.17

Table 2: mAP score (%) of our method (without end-to-end training) as a function of λ_y, λ_f (with $k = 5$) and function of k on the Open Images validation set.

table shows the robustness of our method with respect to k in the Joint Nonnegative OMP algorithm.

Effect of External Knowledge. While we use the label co-occurrence information for building the structure of the label dependency graph, it is important to investigate whether we could achieve improvement by using external knowledge, such as data on the web or WordNet, when available. Thus, we study two alternative approaches.

First, we use Wikipedia to build the structure of the label graph (we still learn its weights using our method). We build the label graph by picking the 50 most frequent concepts in the intro section of the wikipedia article of each label. Since we extract the labels from the web without supervision, our label graph often contain noisy connections. However, our method can learn to remove bad connections by changing the weights of the graph.

Second, we combine Wikipedia and WordNet [53], which is a lexical database for the English language, containing 155,327 words organized in 175,979 synsets. If a label is in the WordNet, we compute the similarities between the word and others using WUP similarity [54] and pick the top 50 similar words as neighbors (the results did not change for similar values). When a label is not in the WordNet, we use the Wikipedia as before.

Table 3 shows the results on the test set of Open Images without representation learning. Notice that the performances of all approaches are similar, only differing by less than 0.02% on, when using all labels. However, wiki

Groups	1	2	3	4	5	All
wiki	69.79	71.35	**76.03**	80.22	**86.32**	**76.74**
wiki+wordnet	69.81	71.17	76.02	**80.27**	86.24	76.72
co-occurrence	**69.91**	**71.36**	75.94	80.15	86.23	76.72

Table 3: mAP score (%) of our proposed method (without end to end training) on the Open Images test set, for using wikipedia vs using wikipedia+wordnet vs estimating co-occurrence from data itself for building the label graph.

performs slightly better than wiki+wordnet. This comes from the fact that similarities in WordNet do not reflect co-occurrences of labels in real images. For example, 'dog' and 'cat', which less frequently co-occur in images, have higher similarity according to the WordNet than 'dog' and 'human', which co-occur in many images. For our co-occurrence label graph, we observe high performance in classes with least annotations, since extracting information from image labels is less noisy than from the web for these classes. Overall, the results show that our co-occurrence method for building the graph is as effective as using external noisy knowledge on the web. On the other hand, as Table 4 showed, not using co-occurrence label graph and fixing its weights does not do as well as using it.

Ablation Study. Table 4 shows the ablation study results of our method by fixing or removing different components. Since labels that have few annotated images also have few testing images, which makes the mAP improvement less statistically meaningful, we report the performance on group 5 that has the most annotated images. Notice that with fixed or without similarity graph, our method performs on par with Curriculum Labeling (85.91%) [3], which shows the importance of our interactive learning scheme. Using a fixed noisy label graph without refinement gives low performance due to the noisy nature of connections learned from limited labeled data. Finally, interactively learning on both image and label graphs (with both similarities being learned) obtains the best performance across different graph construction strategies. As the last row shows,

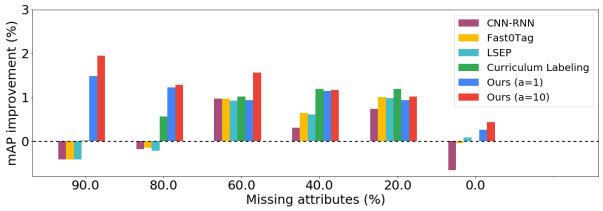

Figure 3: mAP improvement (%) as a function of the percentage of missing attributes in the CUB dataset.

Image Similarity	Label Similarity	mAP
Not used	Not used	85.49
Fixed	Learned	85.83
Learned	Fixed	85.99
Learned	Learned (co-occurrence)	*86.26*
Learned	Learned (wiki)	**86.32**

Table 4: Ablation study on the Open Images dataset.

label graph can embed external knowledge into the learning phase, which performs slightly better than co-occurrence.

Qualitative Results. Figure 2 shows qualitative results on the test set. Our method can capture small objects in images such as Mircrophone, Surfboard or even Hair, thanks to using related labels of semantically similar images. However, our method could face difficulty finding abstract concepts such as Grandparent or Musician. We conjecture such labels depended on the context of an image itself and are hard to transfer based on image similarity alone.

4.6. Results on CUB Dataset

To systematically evaluate the performance of our framework as a function of the percentage of missing labels in all images, we consider the problem of attribute prediction. We experiment on the CUB dataset, which is a fine-grained image dataset of 200 different bird species. Each image in the dataset has an 312-dimensional binary attribute vector.

We select ρ fraction of attributes in each image uniformly at random and drop their values to generate missing attributes. We use our proposed framework to learn attribute classifiers to predict missing attributes in images. To investigate the effect of using images from the same class, we take each partially observed attribute vector and concatenate it with a one-hot encoding vector of the associated class label, whose magnitude of its nonzero element is a. This will only be used on our smoothness loss ℓ_y, defined in (5). A larger a favors selecting similar images from the same class via the similarity learner. This is an advantage of our method that easily incorporates side information, which is not straightforward in other methods. We set the label similarity to identity since attributes are often independent.

Figure 3 shows the mAP scores improvement of different methods over the Logistic method for attribute prediction as a function of different percentage of missing annotations (for clarity, we do not show Latent Noise and Wsabie, which performed worse than other baselines). Notice that with 90% missing attributes, our method achieves about two percent higher mAP score compared to other methods. CNN-RNN treats missing annotations as absent labels which results in poor performance for large fraction of missing attributes. As the percentage of observed attributes increases, the gap in the performance of methods decreases. In general, we observe that our framework does well with large number of missing attributes, thanks to the manifold regularization which is crucial to prevent overfitting (see supplementary material for more detailed results). Finally, our framework with $a = 10$ performs better than $a = 1$, which shows that using images from the same class for attribute learning leads to more accurate results.

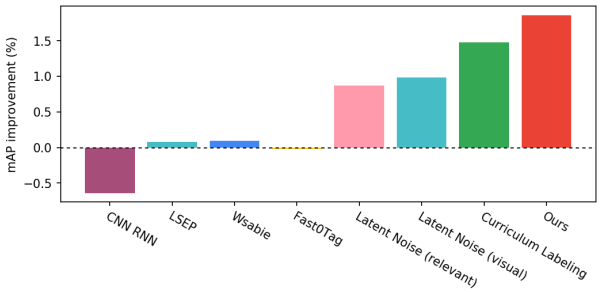

Figure 4: Improvement of mAP score (%) of different methods with respect to the logistic regression on the MS-COCO dataset.

4.7. Results on MS-COCO Dataset

Figure 4 shows the improvement of the mAP score of different methods with respect to the logistic regression baseline. We observe that all methods that can deal with partial labels have significant gain over the logistic baseline while methods that require clean labels have no significant improvement. Moreover, CNN-RNN has low performance even compared to logistic as it treats missing labels as negatives. This demonstrates that limited and noisy annotations are not sufficient to learn good classifiers. Notice that our method outperforms Curriculum Labeling and Latent Noise by 0.38% and 0.88% respectively.

5. Conclusion

We addressed the problem of efficient end-to-end multi-label CNN learning with partial labels on large-scale data. We developed an interactive learning framework that consists of a multi-label CNN classifier and an adaptive similarity learning component that interact and improve the performance of each other. By extensive experiments on the large-scale Open Images dataset as well as CUB and MS-COCO dataset, we showed that our framework improves the state of the art in multi-label learning with partial labels.

Acknowledgements

This work is partially supported by DARPA Young Faculty Award (D18AP00050), NSF (IIS-1657197), ONR (N000141812132) and ARO (W911NF1810300).

References

[1] D. Huynh and E. Elhamifar, "A shared multi-attention framework for multi-label zero-shot learning," *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1

[2] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "Cnn-rnn: A unified framework for multi-label image classification," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2, 5

[3] T. Durand, N. Mehrasa, and G. Mori, "Learning a deep convnet for multi-label classification with partial labels," *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 5, 7

[4] J. Deng, O. Russakovsky, J. Krause, M. S. Bernstein, A. Berg, and L. Fei-Fei, "Scalable multi-label annotation," *SIGCHI Conference on Human Factors in Computing Systems*, 2014. 1, 2

[5] X. Chen, H. Fang, T. Y. Lin, R. Vedantam, S. Gupta, P. Dollar, and C. L. Zitnick, "Microsoft coco captions: Data collection and evaluation server," *arXiv preprint, arXiv:1504.00325*, 2015. 1, 5

[6] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L. J. Li, "Yfcc100m: The new data in multimedia research," *Communications of the ACM*, 2016. 1

[7] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari., "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *International Journal of Computer Vision*, 2016. 1, 5

[8] S. Behpour, W. Xing, and B. D. Ziebart, "Arc: Adversarial robust cuts for semi-supervised and multi-label classification," *AAAI Conference on Artificial Intelligence*, 2018. 1, 2

[9] Y. Guo and S. Gu, "Multi-label classification using conditional dependency networks," *International Joint Conference on Artificial Intelligence*, 2011. 1, 2

[10] X. Xue, W. Zhang, J. Zhang, B. Wu, J. Fan, and Y. Lu, "Correlative multi-label multi-instance image annotation," *International Conference on Computer Vision*, 2011. 1, 2

[11] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten, "Exploring the limits of weakly supervised pretraining," *European Conference on Computer Vision*, 2018. 1, 2

[12] Y. Y. Sun, Y. Zhang, and Z. H. Zhou, "Multi-label learning with weak label," *AAAI Conference on Artificial Intelligence*, 2010. 1, 2

[13] S. S. Bucak, R. Jin, and A. K. Jain, "Multi-label learning with incomplete class assignments," *IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 1, 2

[14] M. Chen, A. Zheng, and K. Weinberger, "Fast image tagging," *International Conference on Machine Learning*, 2013. 1, 2

[15] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," *International Conference on Computer Vision*, 2017. 1, 2

[16] A. Joulin, L. van der Maaten, A. Jabri, and N. Vasilache, "Learning visual features from large weakly supervised data," *European Conference on Computer Vision*, 2016. 1, 2

[17] D. Huynh and E. Elhamifar, "Fine-grained generalized zero-shot learning via dense attribute-based attention," *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1

[18] Y. Gong, Y. Jia, T. Leung, A. Toshev, and S. Ioffe, "Deep convolutional ranking for multilabel image annotation," *International Conference on Learning Representations*, 2013. 1

[19] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid, "Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation," *International Conference on Computer Vision*, 2009. 1

[20] H. F. Yu, P. Jain, P. Kar, and I. S. Dhillon, "Large-scale multi-label learning with missing labels," *International Conference on Machine Learning*, 2014. 1, 2

[21] L. Jing, L. Yang, and J. Y. M. K. Ng, "Semi-supervised low-rank mapping learning for multi-label classification," *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 1

[22] B. Wu, S. Lyu, and B. Ghanem, "Ml-mg: Multi-label learning with missing labels using a mixed graph," *IEEE International Conference on Computer Vision*, 2015. 1, 2

[23] H. Yang, J. T. Zhou, and J. Cai, "Improving multi-label learning with missing labels by structured semantic correlations," *European Conference on Computer Vision*, 2016. 1, 2

[24] Y. Liu, R. Jin, and L. Yang, "Semi-supervised multi-label learning by constrained non-negative matrix factorization," *AAAI Conference on Artificial Intelligence*, 2006. 1, 2

[25] F. Zhao and Y. Guo, "Semi-supervised multi-label learning with incomplete labels," *International Joint Conference on Artificial Intelligence*, 2015. 1

[26] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *Intenational Journal Data Warehousing and Mining*, vol. 3, 2007. 2

[27] Q. Wang, B. Shen, S. Wang, L. Li, and L. Si, "Binary codes embedding for fast image tagging with incomplete labels," *European Conference on Computer Vision*, 2014. 2

[28] A. Veit, N. Alldrin, I. K. G. Chechik, A. Gupta, and S. Belongie, "Learning from noisy large-scale datasets with minimal supervision," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2, 6

[29] H. C. Dong, Y. F. Li, and Z. C. Zhou, "Learning from semi-supervised weak-label data," *AAAI Conference on Artificial Intelligence*, 2018. 2

[30] L. Wang, Z. Ding, and Y. Fu, "Adaptive graph guided embedding for multi-label annotation," *International Joint Conference on Artificial Intelligence*, 2018. 2

[31] I. Misra, A. Shrivastava, and M. Hebert, "Watch and learn:semi-supervised learning of object detectors from videos," *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 2

[32] A. Shrivastava, S. Singh, and A. Gupta, "Constrained semi-supervised learning using attributes and comparative attributes," *European Conference on Computer Vision*, 2012. 2

[33] X. Zhu, "Semi-supervised learning literature survey," *Technical Report 1530, Department of Computer Sciences, University of Wisconsin, Madison*, 2005. 2

[34] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-supervised self-training of object detection models," *IEEE Workshops on Applications of Computer Vision*, 2005. 2

[35] D. Vasisht, A. Damianou, M. Varma, and A. Kapoor, "Active learning for sparse bayesian multilabel classification," *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014. 2

[36] A. Kapoor, R. Viswanathan, and P. Jain, "Multilabel classification using bayesian compressed sensing," *Advances in Neural Information Processing Systems*, 2012. 2

[37] H. M. Chu, C. K. Yeh, and Y. C. F. Wang, "Deep generative models for weakly-supervised multi-label classification," *European Conference on Computer Vision*, 2018. 2

[38] I. Misra, C. L. Zitnick, M. Mitchell, and R. Girshick, "Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels," *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2, 5

[39] Y. Wang and E. Elhamifar, "High-rank matrix completion with side information," *AAAI Conference on Artificial Intelligence*, 2018. 2

[40] R. S. Cabral, F. Torre, J. P. Costeira, and A. Bernardino, "Matrix completion for multi-label image classification," *Advances in Neural Information Processing Systems*, 2011. 2

[41] M. Xu, R. Jin, and Z. H. Zhou, "Speedup matrix completion with side information: Application to multi-label learning," *Advances in Neural Information Processing Systems*, 2013. 2

[42] M. K. Xie and S. J. Huang, "Partial multi-label learning," *AAAI Conference on Artificial Intelligence*, 2018. 3

[43] J. P. Fang and M. L. Zhang, "Partial multi-label learning via credible label elicitation," *AAAI Conference on Artificial Intelligence*, 2019. 3

[44] J. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Information Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004. 4

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 4

[46] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *International Conference on Learning Representations*, 2015. 4

[47] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011. 5

[48] Y. Xian, B. Schiele, and Z. Akata, "Zero-shot learning the good, the bad and the ugly," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 5

[49] J. Weston, S. Bengio, and N. Usunier, "Wsabie: Scaling up to large vocabulary image annotation," *IJCAI*, 2011. 5

[50] Y. Zhang, B. Gong, and M. Shah, "Fast zero-shot image tagging," *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 5

[51] Y. Li, Y. Song, and J. Luo, "Improving pairwise ranking for multi-label image classification," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 5

[52] T. Tijmen and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning 4.2*, 2012. 6

[53] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, 1995. 7

[54] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," *Annual Meeting on Association for Computational Linguistics*, 1994. 7