This WACV 2022 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Meta-Learning for Multi-Label Few-Shot Classification

Christian Simon[†,§], Piotr Koniusz[§,†], Mehrtash Harandi[♣,§]

[†]The Australian National University [♣]Monash University [§]Data61-CSIRO

firstname.lastname@{anu.edu.au,monash.edu,data61.csiro.au}

Abstract

Even with the luxury of having abundant data, multi-label classification is widely known to be a challenging task to address. This work targets the problem of multi-label meta-learning, where a model learns to predict multiple labels within a query (e.g., *an image*) *by just observing a few supporting examples. In doing so, we first propose a benchmark for Few-Shot Learning (FSL) with multiple labels per sample. Next, we discuss and extend several solutions specifically designed to address the conventional and single-label FSL, to work in the multi-label regime. Lastly, we introduce a neural module to estimate the label count of a given sample by exploiting the relational inference. We will show empirically the benefit of the label count module, the label propagation algorithm, and the extensions of conventional FSL methods on three challenging datasets, namely MS-COCO, iMaterialist, and Open MIC. Overall, our thorough experiments suggest that the proposed label-propagation algorithm in conjunction with the neural label count module (NLC) shall be considered as the method of choice.*

1. Introduction

Humans are able to learn novel concepts from very few examples [13, 20]. Our capability of learning from limited examples is attributed to exploiting *background knowledge* [13]. Such knowledge is presented as a collection of relationships, meaning that we are able to learn when various concepts are related and presented together. For example, we can identify a new species of fish, if its relation to concepts such as the sea and other aquatic plants and animals is explained. If the new species merely appears in the jungle among terrestrial animals, humans may fail to identify it without knowing how it relates to other animals.

In machine learning, the so-called domain adaptation [5, 9, 17, 18, 34, 50], zero-shot learning [26, 42, 43], unsupervised/contrastive learning [48, 54, 55, 57], and few-shot learning [11, 12, 19, 27, 30, 31, 35, 40, 41, 44–47, 51, 56] help learning novel concepts from limited data. New samples can also be hallucinated with GANs [6, 28, 29, 49]. Despite the success, the conventional form of episodic learning comes with a limiting assumption of being a single-label problem. In other words, though multi-class, each data sample can only belong to one of possible classes. In the case of images, this means that each image should just encapsulate one visual concept (*i.e.*, object). This clearly limits the application of the developed techniques in many places (*e.g.*, fashion recognition [14], multimedia content analysis [24, 25], bioinformatics [2, 3], and drug discovery [8] to name a few). We bridges this gap by introducing techniques to address Multi-Label Few Shot Learning (ML-FSL).

To get a feeling about the difficulty of ML-FSL, we recall that a profound idea in single-label FSL is to make use of relational comparisons. In essence, one learns to measure the similarity between pairs of samples, where a pair constitutes of the query and an example from the supporting set. While comparing samples in the case of single-labels is well-behaved, extension to the multi-label regime is not an easy ride (see Fig. 1 for a conceptual diagram). This is because, by merely inspecting the similarity between pair of samples, one cannot deduce the class labels of the query. In this work, we extend and introduce novel methods to tackle the problem of ML-FSL. In particular, we investigate:

1. Multi-Label Prototypical Networks. Inspired by the work of Snell *et al.* [32], we make use of the notion of class prototypes to tackle the problem of ML-FSL.

2. Multi-Label Relation Networks. Building upon the work of Sung *et al.* [33], we introduce multi-label relational networks with binary relevance.

3. Label Propagation Networks. We propose a label propagation algorithm to address the problem of ML-FSL. Label propagation models the data in the form of graphs. This model learns the correlations among samples by weighting mechanism according to their similarities.

To the best of our knowledge, only the work of Alfassy *et al.* [1] tackles the problem of ML-FSL. Our work goes

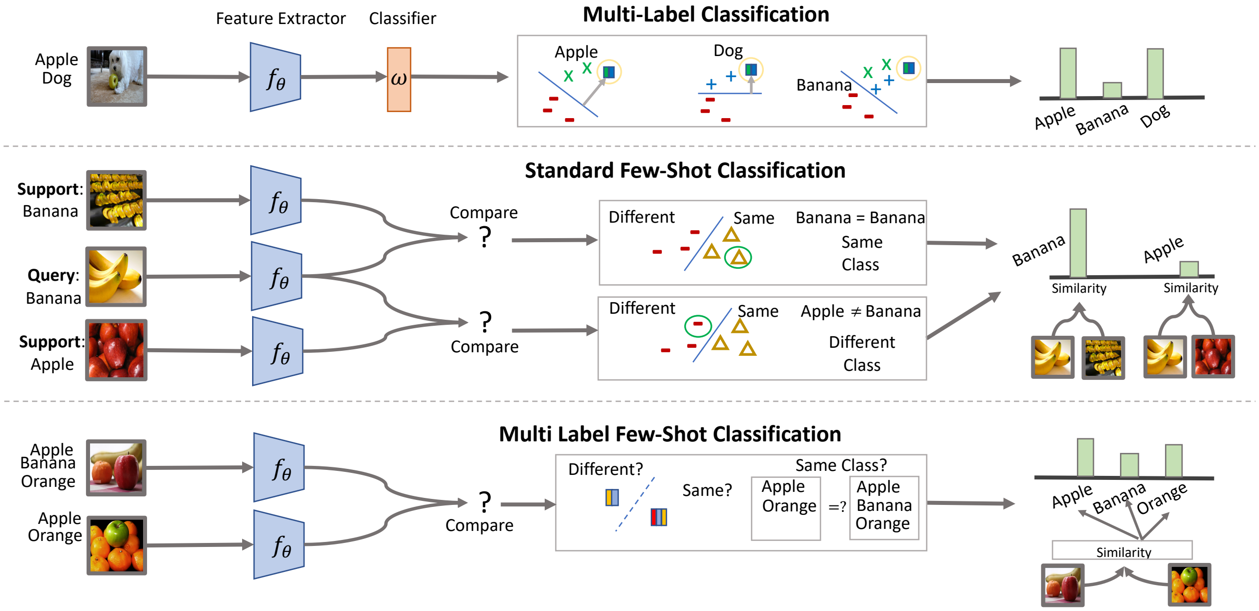

Figure 1. Comparison between multi-label classification, single-label few-shot learning and multi-label few shot learning. Multi-label few-shot problem. **Top panel.** In classical multi-label classification, one designs a fixed classifier from all seen classes. In a well-established practice, one breaks down the problem into identifying a set of binary classifiers where each classifier is responsible for identifying one specific class in a given input. **Middle panel.** A profound idea in addressing single-label FSL is to design a model to perform relational comparisons. Here, the model will receive pairs of images (the query and an image from the support set) and predicts whether they are similar or not. We note that while in multi-label problems, the embedding of the query image is fixed, in relational inference, the embedding is dependent on the constructed pairs. **Bottom panel.** Extension of the relational inference to multi-label FSL regime is not trivial. Our work provides various solutions to address this challenging, yet extremely important and practical problem.

beyond this work in various ways. This includes, extending and introducing new algorithms for ML-FSL, developing a comprehensive evaluation framework with three challenging datasets, namely MS-COCO [22], iMaterialist [7], and Open MIC [18] and proposing a neural model to perform label count, a module that empirically seems to be boosting the accuracy by a tangible margin. To summarize, our contributions in this paper are:

i. By departing from single-label FSL problems, we generalize and introduce various techniques to address the ML-FSL problem.

ii. A comprehensive and challenging evaluation framework for ML-FSL is developed.

iii. We propose a neural label count module (NLC) to estimate the number of labels in an input and thoroughly asses its efficiency in conjunction with the proposed ML-FSL solutions.

2. Problem Definition

Our goal is to construct meta-learning tasks such that the models can gain experience or learn from similar tasks. Inspired by [35], we propose the concept of multi-label few-shot classification via learning from *episodes*. Moreover, the finite set estimation (label count) is also considered in this setting to complement the mAP measurement. Below, we explain our setting, notations, training, and testing strategies for ML-FSL.

Notations. Throughout this paper, we use bold lower-case letters (*e.g.*, $\boldsymbol{x}$) to denote column vectors and bold upper-case letters (*e.g.*, $\boldsymbol{X}$) to denote matrices. A network is denoted by $f_\Theta : \mathbb{R}^m \to \mathbb{R}^n$ which maps an input into some feature space. The set of labels per *episode* is represented by $\mathcal{C}$. Let $\mathcal{X} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_{N_x}, \boldsymbol{y}_{N_x}))\}$ with $\boldsymbol{x}_i \in \mathbb{R}^n; \boldsymbol{y}_i \in \mathbb{R}^{|\mathcal{C}|}$ and $\mathcal{Q} = \{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_{N_q}\}$ with $\boldsymbol{q}_i \in \mathbb{R}^n$ denote the support set and query set, respectively. Element-wise index in a vector is denoted as $\boldsymbol{x}_{(j)}$.

Episode. To form an episode (see Fig. 2), we sample from a task distribution T over possible label sets. There are two sets shaping an *episode*: a support set $\mathcal{X}$ and a query set $\mathcal{Q}$. To address the multi-label classification, $\boldsymbol{y}_i \subseteq \mathcal{C}$ denotes a set of multiple labels assigned to $\boldsymbol{x}_i$. Note that, an episode composition including selected classes and examples may differ from one episode at a given timestep to another. This *episode* structure exploits meta-knowledge such that a model that has the meta-learning capability will learn to match representations regardless of the actual semantic meaning of

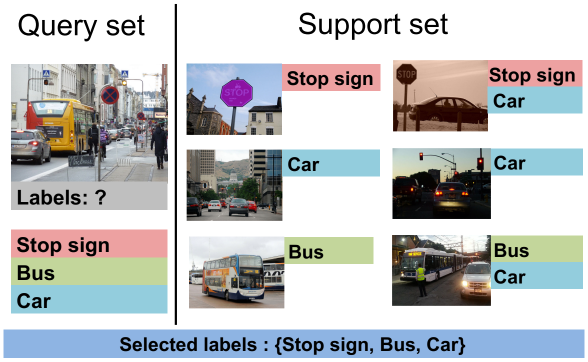

Figure 2. An episode consists of a query set and a support set. The support set contains examples from selected classes of a given task. The query set covers the same set of labels presented in the support set.

the labels, thus learning a relation between data points. As a result, the models can generalize and predict the appropriate labels given only a few data points during training. The challenges such a technique has to address are twofold: (i) the number of data points is low and (ii) the size of the label set varies from image to image thus making the prediction task harder.

N-way K-shot. In single-label few-shot learning, the term N-way K-shot is used to describe the number of classes (way) in an episode and the number of examples (shot) in each class. For example, suppose there are three sampled classes ($N = 3$), then we obtain five examples ($K = 5$) per class to compose an *episode*, so the support set contains 15 images in total. Following this notation, we also sample N-way and K-shot from each label to form an *episode* in ML-FSL. However, the support set composition for ML-FSL differs from the single-label setting in the sense that the distribution over samples per class in an episode is not uniform. Because each data point $\boldsymbol{x}_i$ in a support set can contain more than one label from the set of labels $\mathcal{C}$, thus, it is not guaranteed that there are exactly $K \times N$ samples constituting the support set of an episode.

Training Stage. In majority of cases, to perform multi-label classification, pre-trained CNNs are used [36, 37, 53]. However, in this paper, we want to focus on the learning techniques for which the network must exhibit the capacity to improve its performance by learning from previous tasks. As a result, we train the networks from scratch by random initialization. The models are updated based on the training objective as follows:

$$\Theta^* = \arg\max_\Theta \mathbb{E}_{\mathcal{C}\sim T}\left[\mathbb{E}_{\mathcal{X}\sim\mathcal{C},\mathcal{Q}\sim\mathcal{C}}\left[\sum_{\boldsymbol{q}\in\mathcal{Q}} \log P_\Theta(\boldsymbol{y}|\boldsymbol{q},\mathcal{X})\right]\right], \quad (1)$$

where Θ denote the model parameters that we want to learn. The model is trained over many episodes to minimize the prediction error over the query set Q.

Testing Stage. In the testing stage, the model performs classification via assigning each data point to unseen labels from a distribution T'. Following an *episode* composition in the training stage, there are N labels and at least K examples per label in the support set. Thus, one can think of such classification strategy as transfer learning from previously learned tasks to the new set of tasks.

3. Meta-Learning for Multi-Label Few-Shot Problems

In this section, we first extend three successful FSL schemes to their ML-FSL counterparts (see Fig. 3 for a conceptual diagram). This is followed by describing the proposed NLC module.

3.1. From Single-Label to Multi-Label

Below, we extend and adapt prototypical networks [32], relation networks [33], and label propagation [23] to their multi-label versions.

Prototypical Networks. Prototypical networks learn a mapping $f_\Theta : \mathbb{R}^m \to \mathbb{R}^n$ from an input space to an embedding space. The main assumption is that the embeddings should sit near their class prototypes. Let us denote the prototypes (for C classes in an episode) by $\boldsymbol{P} = \{\boldsymbol{p}_1, \dots, \boldsymbol{p}_C\}; \boldsymbol{p}_k \in \mathbb{R}^n$ and $\boldsymbol{X}_i$ be a class-specific set. A prototype is calculated as the mean of the feature vectors that share the same class label:

$$\boldsymbol{p}_k = \frac{1}{|C_k|} \sum_{\boldsymbol{x}_j \in \boldsymbol{X}_k} f_\Theta(\boldsymbol{x}_j). \quad (2)$$

In prototypical networks, a query feature is classified by assigning it to the nearest prototype according to the Euclidean distance: $\boldsymbol{z}_{(j)} = \|\boldsymbol{p}_j - f_\Theta(\boldsymbol{q})\|^2$.

We adapt prototypical networks to work in the multi-label setting. For a given class, all samples annotated with that class are grouped together to form a prototype. Note that every sample with more than one label will contribute in formation of several prototypes. To perform a multi-label classification, we conduct training with a softmax function [37, 39] in the final layer of network. The objective function is formulated as follows:

$$\mathcal{L}_{PN} = \sum_{j=1}^{|\mathcal{C}|} \left(\frac{\boldsymbol{y}_{(j)}}{\|\boldsymbol{y}\|_1} - \frac{\exp(-\boldsymbol{z}_{(j)})}{\sum_{j'=1}^{|\mathcal{C}|} \exp(-\boldsymbol{z}_{(j')})} \right)^2. \quad (3)$$

Relation Networks. Few-shot recognition can also be performed via the use of the so-called *relation module* from relation networks (RN). This approach can be viewed as learning deep non-linear metric. By and large, RN consists of an embedding module and a *relation module*. Specifically, an embedding module is a non-linear mapping from the input space to a feature space $f_\Theta : \mathbb{R}^m \to \mathbb{R}^n$ and the *relation module* (g_Φ) learns the similarity between the query ($\boldsymbol{q}$) and sample ($\boldsymbol{x}_j$) in the support set. Let us denote $\boldsymbol{X}_i$

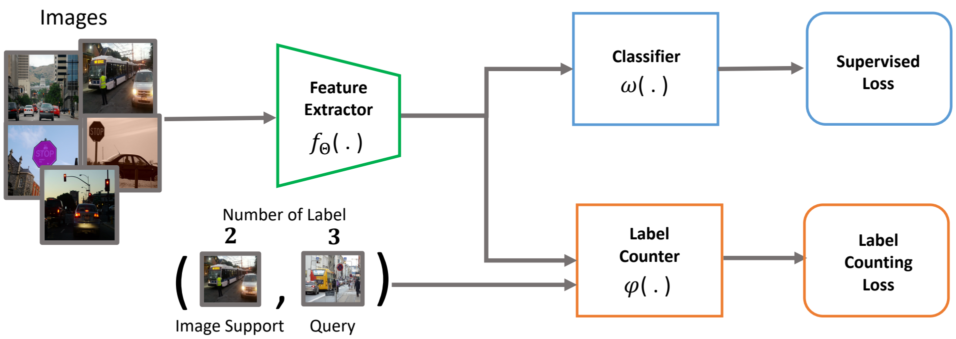

Figure 3. The general architecture to classify and predict the number of labels in training.

be a class-specific set. Given a problem with $\mathcal{C}$ classes, the relation scores can be calculated according to:

$$\boldsymbol{r}_{(j)} = g_\Phi\Big(f_\Theta(\boldsymbol{q}), \frac{1}{|C_j|}\sum_{\boldsymbol{x}_i \in \boldsymbol{X}_j} f_\Theta(\boldsymbol{x}_i)\Big), \quad j = 1, \ldots, \mathcal{C}. \tag{4}$$

Architecture-wise, the *relation module* is comprised of two convolutional blocks, two fully connected layers, and a sigmoid function (σ). Training is performed by calculating mean squared error between the score and the query label w.r.t. the model parameters and embeddings:

$$(\Theta, \Phi) = \arg\min_{\Theta,\Phi} \sum_{j=1}^{|\mathcal{C}|} (\boldsymbol{r}_{(j)} - \boldsymbol{y}_{(j)})^2. \tag{5}$$

Thus, RN passes the features from the support set and a query to the binary classifier and the label with highest score is chosen as the predicted class.

Adapting RN from single-label to multi-label setting is straightforward. The *relation module* acts as a non-linear metric, thus, we can directly use a log-loss w.r.t. $\boldsymbol{r}_{(j)}$ and labels y_j:

$$\mathcal{L}_{RN} = \sum_{j=1}^{|C|} \boldsymbol{y}_{(j)} \log(\boldsymbol{r}_{(j)}) + (1 - \boldsymbol{y}_{(j)}) \log(1 - \boldsymbol{r}_{(j)}). \tag{6}$$

Label Propagation. Another approach is to model few-shot learning with a graph and make use of label propagation methods to classify a query sample. To predict the concept scores, we make use of the ***smoothness*** property, meaning that similar instances should have similar concept scores. More specifically, if a query sample $\boldsymbol{q}_j$ is similar to a support instance $\boldsymbol{x}_i$, then $\boldsymbol{\psi}_j \approx \nu(\boldsymbol{y}_i)$ where $\nu : \mathbb{R}^C \to \mathbb{R}^C$ is the ℓ_1 normalization operator defined as $\nu(\boldsymbol{y}) = \boldsymbol{y}/\|\boldsymbol{y}\|_1$. Let $\mathbb{R}^{n \times (N_s + N_q)} \ni \boldsymbol{X} = [\mathcal{X}, \mathcal{Q}] = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{N_s}, \boldsymbol{q}_1, \cdots, \boldsymbol{q}_{N_q}]$. We denote the columns of $\boldsymbol{X}$ with $\boldsymbol{x}_i; 1 \leq i \leq N_s + N_q$. Similarly, define $\mathbb{R}^{C \times (N_s + N_q)} \ni \boldsymbol{\Phi} = [\nu(\boldsymbol{y}_1), \nu(\boldsymbol{y}_2), \cdots, \nu(\boldsymbol{y}_{N_s}), \boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \cdots, \boldsymbol{\psi}_{N_q}]$. $\boldsymbol{\Phi}_\mathcal{X}$ has a similar structure with $\boldsymbol{\psi}_i = 0, i = N_s + 1, \cdots, N_s + N_q$.

To achieve our goal, we start by building a neighborhood graph (W) over the support and the query sets using $\boldsymbol{X}$ according to

$$w_{i,j} = \begin{cases} \exp\big(-\sigma\|f_\Theta(\boldsymbol{x}_i) - f_\Theta(\boldsymbol{x}_j)\|^2\big), & j \in \mathcal{N}_i \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Here, $\mathcal{N}_i$ is the set of neighbors of node i that share the same class label to $\boldsymbol{x}_i$. Then, a normalized graph is formed with $\boldsymbol{S} = \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{W} \boldsymbol{D}^{-\frac{1}{2}}$ where $\boldsymbol{D}$ is a diagonal matrix with $\boldsymbol{D}_{ii} = \sum_j \boldsymbol{W}_{ij}$. Then final prediction scores has a closed form (see [52] for example) and can be calculated:

$$\boldsymbol{F}^* = \boldsymbol{\Phi}_\mathcal{X}(\mathbf{I} - \alpha\boldsymbol{S})^{-1}, \tag{8}$$

where α is usually set to 0.99 in practice. The final classification loss for multi-label setting is defined as:

$$\mathcal{L}_{LP} = \big\|\boldsymbol{\Phi} - \boldsymbol{F}^*\big\|^2. \tag{9}$$

3.2. Neural Label Count

We begin with the role of self-supervision learning for the multi-label problem. The characteristic of self-supervision training is to use the characteristic of the data. For instance, by rotating an image, one can self-supervise a machine by predicting the amount of rotation. Here, we make use of the arithmetic operation for prediction. The functionality of the NLC module is self-explanatory. The module predicts the number of classes (*e.g.*, objects in an image) presented in a given input. Our design benefits from a relation module that takes into account a support sample, the query, and the global information of a task, represented in the support set . The function to predict the number of labels is denoted by:

$$M^{[\boldsymbol{x_i}, \boldsymbol{q}]} = \varphi(f_\Theta(\boldsymbol{x_i}), f_\Theta(\boldsymbol{q}), \boldsymbol{z}), \tag{10}$$

where the multi-label counter function $\varphi : \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{2|C|}$ learns the relation between two samples (*i.e.*, $f_\Theta(\boldsymbol{x_i})$

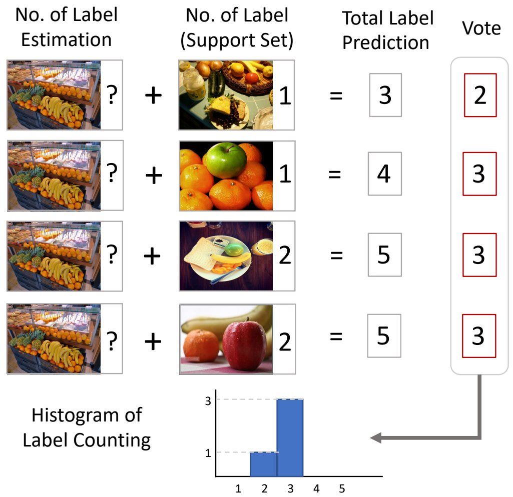

Figure 4. Estimating the number of label by voting system of the support samples and queries. Here, an estimation (top) predicts the number of label as 2 but the rest predictions are 3.

and $f_\Theta(\boldsymbol{q})$). Here, $\boldsymbol{z}$ is a vector carrying the context of the whole support set to the NLC module. In our implementation, we realize this as $\boldsymbol{z} = \frac{1}{NK}\sum_{\boldsymbol{x}_i \in \mathcal{X}} f_\Theta(\boldsymbol{x_i})$.

In particular and as becomes clear shortly, the NLC predicts the number of objects presented collectively in $\boldsymbol{x_i}$ and $\boldsymbol{q}$. In doing so, the NLC module uses a softmax with $2|C|$ outputs as the maximum number of classes presented collectively in $\boldsymbol{x_i}$ and $\boldsymbol{q}$ cannot exceed $2|C|$.

Our training OBJECTIVE is to minimize the following loss function:

$$\mathcal{L}_E = \mathcal{L}_{su} + \lambda \mathcal{L}_{co}, \tag{11}$$

where $\mathcal{L}_{su}$ depends on the selection of the classifier and $\mathcal{L}_{co}$ is formulated as:

$$\mathcal{L}_{co} = -\sum_j \sum_i \log\left(\frac{\exp(M_{(j)}^{[\boldsymbol{x}_i,\boldsymbol{q}]})}{\sum_{j'} \exp(M_{(j')}^{[\boldsymbol{x}_i,\boldsymbol{q}]})}\right). \tag{12}$$

3.3. Inference with Label Count Voting

We employ a voting scheme on the output of the NLC module. To be more specific, we compare each support sample with the query and create the histogram of the label count estimation as shown in Fig. 4. The label counting based on histogram ($\boldsymbol{H}$) is defined as:

$$\boldsymbol{H} = \{h(m) | 0 \leq m \leq 2|C|\}, \tag{13}$$

with

$$h(m) = \sum_{\boldsymbol{x}_i \in \mathcal{X}} \delta((M^{[\boldsymbol{x}_i,\boldsymbol{q}]} - B^{\boldsymbol{x}_i}) - m), \tag{14}$$

where $B^{\boldsymbol{x}_i}$ is the label count of the support sample $\boldsymbol{x}_i$ and $\delta(\cdot)$ returns 1 if its argument is equal 0 (or it returns 0 for the input not equal 0). Indeed, $h(m)$ shows how many times, the number of classes in the query is counted as m. A majority voting is then used to make the final call as:

$$l = \underset{i \in \{1,\cdots,|C|\}}{\arg\max} \; h(i). \tag{15}$$

This label count estimation is also included in our evaluation scheme.

Remark 1 *We consider the episode style for training and testing multi-label few-shot classification. In this setting, the labels per episode are randomly picked and every episode may contain different samples. Thus, there is a chance that some objects are labeled in an episode but they are not labeled in another episode. This is also known as missing labels in multi-label classification. Based on this problem, the relation module is designed to be an adaptive module that capture the information based on the context in an episode.*

Remark 2 *The NLC module has a different use compared to relation networks [33] that exploits the similarity between two features. The neural label count module conveys the operator relationship (*i.e.*, summation) such that the result of this relation module is the number after an arithmetic operation.*

Remark 3 *This inference method makes use of the ensemble strategy to estimate the label count. The ensemble of label count estimations implies that a level of robustness might be expected. That is, even if one of the classifiers is wrong in its prediction, the other predictions may correct it through consensus. The final prediction is made based on the most frequent label count predictions. In the case of a tie, then the original predictions (floating point) are used to check the number with the highest probability.*

4. Related Work

In the multi-label setting, the main challenge is to classify an example belonging to multiple different classes simultaneously. The problem is challenging as an algorithm has to return a correct set of labels for previously unseen sample (the label count differs per sample).The problem at hand is obviously more difficult as the solution needs to infer not only the most likely labels but also decide on their numbers.

Few-shot Learning. Metric-learning approaches are proven beneficial to tackle classification tasks with low number of samples [16, 32, 33]. The embedding networks learn to compare contents of episodes in order to infer the underlying discriminative model for the given tasks. Initially, siamese [16] and matching networks [35] apply the idea of nearest neighbor inference for the task of few-shot classification. Prototypical networks [32] models the feature representations from samples within the same class as a single prototype. Furthermore, a relationship between class

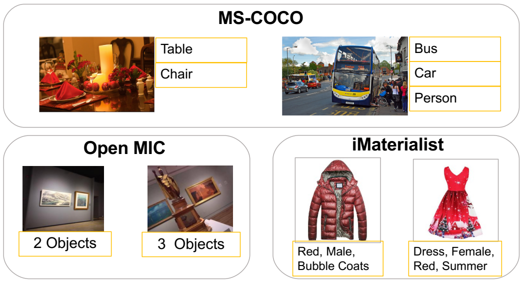

Figure 5. Sample images from three datasets: MS-COCO, iMaterialist, and Open MIC.

representations and queries can be also learnt through a neural network unit *e.g.*, so-called Relation Networks [33]. Such a family of few-shot learning techniques learns an underlying metric, that is, the distance between any two feature vectors becomes small only if they belong to the same class. Another few-shot model using metric learning is through model adaptation as proposed in [10, 31]. Another work uses graph as transductive label propagation networks in [23]. These works are related to our extension from single-label few-shot learning to ML-FSL.

Multi-label Classification. Focusing on deep learning solutions for multi-label classification, the majority of studies use the so-called joint embedding models to solve the task in-hand [4, 38]. The multi-label classification problem can also be tackled in a sequential manner via a recurrent neural network (RNN) [36] which aggregates feature vectors produced by a CNN feature encoder. Recently, attention-based approaches for multi-label image recognition have been developed. An iterative attentional regions discovery is proposed in [37]. These attentional regions correspond to semantic labels and thus they are embedded via LSTM. Moreover, a spatial regularization network (SRN) [53] learns semantic and spatial label relations from attention maps. Thus, initial confidence scores can be adjusted via SRN to new regularized scores. The correlation between two labels and an image feature obtained from CNN is expressed via correlation matrix. Then, the determinant of the correct subset is maximized to learn the deep neural network (DNN). Intuitively, one can think of this approach as maximizing a set of subspaces, each spanned by one ground-truth set of labels and corresponding features while minimizing the remaining sets of subspaces. Moreover, a previous work [21] uses pair wise ranking and threshold or label count estimation to improve the result of multi-label classification.

5. Experiments

5.1. Datasets

We use three datasets detailed below to evaluate the capability of the methods to conduct ML-FSL. Note that, we have two disjoint sets for training and testing, so the images with annotated classes in the testing set do not appear in the training set. We propose the new splits for these three datasets. Fig. 5 shows examples from all datasets.

Model	1-shot	5-shot
LASO (intersection aug.)	40.5	57.2
LASO (union aug.)	45.3	58.1
Proto Nets	48.7	59.9
Relation Nets	49.5	58.5
LPN	56.1	63.4
Proto Nets + NLC	**50.2**	**60.4**
Relation Nets + NLC	**53.3**	**60.8**
LPN + NLC	**56.8**	**64.8**

Table 1. A comparison in mAP(%) to the existing benchmark [1] on 16-way 1-shot and 5-shot.

MS-COCO [22]. The MS-COCO dataset is built for object detection and recognition, thus, the data is suitable for multi-label recognition. This dataset comprises 80 classes and 122,000 images in total. We split the training, validation, and testing set into 50, 10, and 20 classes, respectively. We categorize 50 training classes as base classes and 20 testing classes as novel classes. All of the images within training are disjoint from validation and testing sets. Additionally, we remove images that have less than two labels yielding 74,655 images. We use all of the images from the validation set of MS-COCO to evaluate performance on base classes.

In addition, we also evaluate on the MS-COCO split in [1] to compare with the existing approach. This split has 64 and 16 classes for training and testing, respectively.

iMaterialist. We evaluate our ML-FSL proposal on the iMaterialist fashion dataset [7] consisting more than 1 million images and 228 distinct labels. We divide the dataset into training, validation, and testing sets for our purpose. We consider the subset of the dataset of 120 labels from all labels. The splits for testing and validation are 40 and 15, respectively. The rest labels are used for training. This fashion dataset shows the multi-label problem that has multiple labels for one object. For example, a color and a texture share the same visual appearance but they have different labels.

Open MIC [18]. This dataset contains images collected from 10 museum exhibition spaces. There are 866 classes in total and every class comprises 1-20 images per class. The images suffer from various photometric and geometric distortions. Multi-label annotations are available as every image contains more than one exhibit. The dataset is split into four subsets: *p1=(shn+hon+clv)*, *p2=(clk+gls+scl)*, *p3=(sci+nat)*, *p4=(shx+rlc)*.

5.2. Details

Implementation. We equip all methods in our experiments with the same variant of convolutional neural networks (CNN). The CNN has 4-convolutional layers (4-Conv) with

Model	Base Classes				Novel Classes			
	1-shot		5-shot		1-shot		5-shot	
	mAP	LC	mAP	LC	mAP	LC	mAP	LC
Pre-trained + MLP	56.6	-	57.5	-	50.2	-	54.4	-
Proto Nets	61.0	-	69.7	-	56.7	-	66.7	-
Relation Nets	64.4	-	69.3	-	57.3	-	63.3	-
LPN	64.8	-	71.8	-	58.5	-	68.3	-
Proto Nets + NLC	62.8 ↑	40.2	72.3 ↑	45.1	58.0 ↑	**49.5**	68.0 ↑	54.1
Relation Nets + NLC	**66.2** ↑	**42.7**	71.0 ↑	42.8	59.0 ↑	48.0	66.1 ↑	**57.0**
LPN + NLC	66.0 ↑	39.0	**74.5** ↑	**46.0**	**60.4** ↑	47.4	**69.1** ↑	54.6

Table 2. The accuracy (%) of baseline methods and the additional NLC on MS-COCO. The evaluation is based on mAP and LC for 10-way 1-shot and 5-shot. The best performance is in **bold** and the second best is in blue font.

Model	MS-COCO				iMaterialist			
	1-shot		5-shot		1-shot		5-shot	
	10-way	15-way	10-way	15-way	10-way	15-way	10-way	15-way
Proto Nets + NLC	30.4	20.9	37.1	24.8	45.6	**43.6**	47.4	**47.2**
Relation Nets + NLC	26.2	14.3	29.2	17.27	**46.5**	42.7	**49.7**	47.0
LPN + NLC	**31.1**	**24.2**	**37.8**	**28.42**	45.4	40.4	48.9	43.5

Table 3. The accuracy (%) which quantifies multi-label hard predictions. The counted label is used in making predictions based on thresholds.

Model	15-way			
	1-shot		5-shot	
	mAP	LC	mAP	LC
Proto Nets	48.4	-	59.8	-
Relation Nets	52.2	-	55.9	-
LPN	52.3	-	61.1	-
Proto Nets + NLC	49.4 ↑	36.5	61.0 ↑	**45.0**
Relation Nets + NLC	**54.4** ↑	**41.1**	54.4 ↑	35.7
LPN + NLC	53.6 ↑	38.1	**63.6** ↑	43.6

Table 4. The accuracy (%) of baseline methods with and without the auxiliary NLC loss on the MS-COCO. The evaluation is based on mAP and LC for 15-way 1-shot and 15-way 5-shot protocols. The best performance is highlighted by the **bold** font and the second best by the blue font.

Model	15-way			
	1-shot		5-shot	
	mAP	LC	mAP	LC
Proto Nets	52.9	-	62.6	-
Relation Nets	58.4	-	64.6	-
LPN	56.4	-	59.2	-
Proto Nets + NLC	54.0 ↑	28.0	64.1 ↑	**31.9**
Relation Nets + NLC	**59.9** ↑	27.7	**65.3** ↑	31.9
LPN + NLC	57.4 ↑	**29.5**	61.0 ↑	29.5

Table 5. The accuracy (%) of baseline methods without and with the auxiliary NLC loss on iMaterialist. The evaluation is based on mAP and LC for the 15-way 1-shot and 15-way 5-shot protocols. The best performance is highlighted by the **bold** font and the second best by the blue font.

Model	1-shot		5-shot	
	mAP	LC	mAP	LC
Proto Nets	60.8	-	66.4	-
Relation Nets	62.1	-	67.4	-
LPN	62.3	-	65.2	-
Proto Nets + NLC	62.6 ↑	32.2	68.9 ↑	35.1
Relation Nets + NLC	**64.0** ↑	39.0	**69.0** ↑	35.7
LPN + NLC	63.5 ↑	**42.4**	66.8 ↑	**37.0**

Table 6. The accuracy (%) of baseline methods and the additional NLC on iMaterialist. The evaluation is based on mAP and LC for 10-way 1-shot and 5-shot. The best performance is in **bold** font and the second best is in blue font.

64 filters in each layer followed by batch normalization, ReLU, and max-pooling. Training and testing is performed over 100,000 and 1,000 episodes for both architectures. Adam optimizer [15] with learning rate 0.001 is used whose learning rate is reduced by half every 10,000 episodes. We use $\lambda = 0.01$ for all methods and datasets.

We perform 10-way 1-shot and 10-way 5-shot experiments to compare the performance. The number of query images is half of the number of sampled labels. On MS-COCO, the testing stage employs a model from training for which the best validation score was attained. On the Open MIC dataset, we employ the models saved on the the last training *episode*.

Evaluation Metric. The evaluation metric to measure accuracy in our experiments is based on mean average precision (mAP) which is the ranked list of the confidence scores compared with the list of the true labels. Another measurement is the accuracy to estimate the label count (LC).

5.3. Results and Ablation Studies

In our experiments, we consider the multi-label classification problem in the meta-learning setting to explore the assumption that the tasks given during training and testing

Model	1-shot				5-shot			
	$p1 \rightarrow p2$	$p2 \rightarrow p3$	$p3 \rightarrow p4$	$p4 \rightarrow p1$	$p1 \rightarrow p2$	$p2 \rightarrow p3$	$p3 \rightarrow p4$	$p4 \rightarrow p1$
Pre-trained + MLP	62.24	52.85	66.89	51.12	83.93	74.93	86.93	70.63
Proto Nets	58.48	54.82	67.41	50.89	81.52	75.71	86.07	72.64
Relation Nets	48.50	48.74	63.03	50.93	53.57	66.26	72.50	57.20
LPN	**65.41**	**60.06**	**74.60**	**53.02**	**91.34**	**79.57**	**90.89**	**77.15**

Table 7. ML-FSL results (mAP) on Open MIC for 10-way 1-shot and 5-shot. $p\{n\} \rightarrow p\{m\}$ means that training is performed in $p\{n\}$ and testing is applied in $p\{m\}$. The best performance is in **bold** font and the second best is in blue font.

are similar. The experiments are performed with baselines on the 4-Conv backbone. In addition to the baselines, we perform a comparison to the pre-trained feature extractor on the training set and a multi-layer perceptron (MLP) layer as a classifier which is updated based on the given support set. The experiments are run on 10-way and 15-way with 1-shot and 5-shot protocols as shown in Table 2, 6, 5 and 4. In addition, we also provide a comparison to the existing ML-FSL method in Table 1.

MS-COCO. In this dataset, we evaluate the methods on the base classes and the novel classes. This protocol makes sure that the model does not only fit to the novel classes and *forget* the base classes. We can observe the results from Table 2 that pre-trained feature extractor with MLP cannot perform better than the few-shot learning baselines trained from scratch with episodic training. In all cases, label propagation network (LPN) can outperform the other methods. LPN outperforms the second highest on novel classes by 1.4% and 1.1% for 1-shot and 5-shot, respectively. This is because a graph model can capture the relation among samples which share similarity. Furthermore, there are also improvements for both base and novel classes when the basic few-shot learning methods are complemented with NLC. We conjecture that NLC imposes a regularization implicitly. On MS-COCO split by [1], we outperform the LASO model by 10% and 6% for 16-way 1-shot and 16-way 5-shot, respectively (see Table 1).

iMaterialist. In this dataset, the data has a different structure compared to MS-COCO and Open MIC because two labels can share the same the same object and the labels are hierarchical. Adding a label count loss is consistently beneficial for improving the accuracy (mAP). We observe that a further improvement $\sim$1.5% with NLC for 10-way 1-shot and 5-shot as shown in Table 6. Relation Nets [33] has the highest performance for 10-way 1-shot and 5-shot. Our conjecture is that the fashion images have less tight relationship between one label to another label than images in MS-COCO. For instance, a red color can appear to any fashion entities such as dress, trouser, or shirt but it is unlikely that an apple can appear on images containing sport equipment.

Open MIC. On Open MIC dataset, the evaluation is performed on 4 different exhibitions for 10-way 1-shot and 5-shot. In all cases, LPN outperforms by significant margins compared to the other three baselines with minimum of 3% in mAP as shown in Table 7.

Hard Decision. Furthermore, we also provide the accuracy scores based on the hard decision formed by the threshold obtained from the neural label count. Table 3 shows that making hard predictions on image contents is very challenging because we need to predict the probability of classes and decide which final labels appear in a given image. This is non-trivial in relational learning where testing classes are disjoint from training classes.

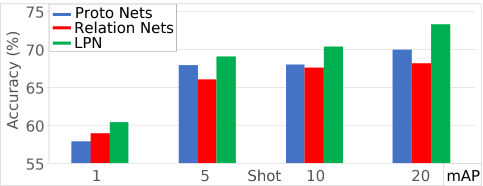

Figure 6. The impact of shot on MS-COCO (mAP).

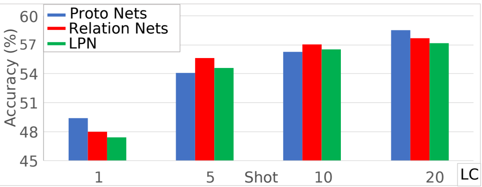

Figure 7. The impact of shot on MS-COCO (LC).

The Impact of Shot. We investigate the effect of the number of samples for mAP and LC. It is clearly shown in Fig. 7 that more additional samples are beneficial to estimate the label count. The baselines are also improved when more data is given as shown in Fig. 6. We can observe that mAP and LC increase about 10% or more from 1-shot to 20-shot.

6. Conclusions

In this paper, we introduce a meta-learning framework for multi-label few-shot classification and a label counting module. The meta-learning is given in the form of *episode* such that the models can gain experience by learning from past tasks and generalize to new tasks which are similar in their nature to the past tasks. To this end, we have measured the performance on three challenging datasets: MS-COCO, iMaterialist, and Open MIC. Apart from multi-label few-shot learning protocols, we have proposed a neural label count module to estimate the number of labels. This is built based on the voting system to handle weak predictions. We showed that the module is beneficial to improve the performance even more.

References

[1] Amit Alfassy, Leonid Karlinsky, Amit Aides, Joseph Shtok, Sivan Harary, Rogerio Feris, Raja Giryes, and Alex M Bronstein. Laso: Label-set operations networks for multi-label few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6548–6557, 2019. 1, 6, 8

[2] Zafer Barutcuoglu, Robert E Schapire, and Olga G Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7):830–836, 2006. 1

[3] Nicolò Cesa-Bianchi, Matteo Re, and Giorgio Valentini. Synergy of multi-label hierarchical ensembles, data fusion, and cost-sensitive methods for gene functional inference. *Machine Learning*, 88(1-2):209–241, 2012. 1

[4] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013. 6

[5] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073, 2012. 1

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Conference on Neural Information Processing Systems*, 2014. 1

[7] Sheng Guo, Weilin Huang, Xiao Zhang, Prasanna Srikhanta, Yin Cui, Yuan Li, Matthew R.Scott, Hartwig Adam, and Serge Belongie. The imaterialist fashion attribute dataset. *arXiv preprint arXiv:1906.05750*, 2019. 2, 6

[8] Dominik Heider, Robin Senge, Weiwei Cheng, and Eyke Hüllermeier. Multilabel classification for exploiting cross-resistance information in hiv-1 drug resistance prediction. *Bioinformatics*, 29(16):1946–1952, 2013. 1

[9] S. Herath, M. Harandi, and F. Porikli. Learning an invariant hilbert space for domain adaptation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1

[10] Jie Hong, Pengfei Fang, Weihao Li, Tong Zhang, Christian Simon, Mehrtash Harandi, and Lars Petersson. Reinforced attention for few-shot learning and beyond. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 913–923, 2021. 6

[11] Huaxi Huang, Junjie Zhang, Litao Yu, Jian Zhang, Qiang Wu, and Chang Xu. TOAN: Target-oriented alignment network for fine-grained image categorization with few labeled samples. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021. 1

[12] Huaxi Huang, Junjie Zhang, Jian Zhang, Jingsong Xu, and Qiang Wu. Low-rank pairwise alignment bilinear network for few-shot fine-grained image classification. *IEEE Transactions on Multimedia*, 23:1666–1680, 2021. 1

[13] T. Hume and M. J. Pazzani. Learning sets of related concepts: A shared task model. In *Proceedings of Annual Conference of the Cognitive Science Society*, 1996. 1

[14] Naoto Inoue, Edgar Simo-Serra, Toshihiko Yamasaki, and Hiroshi Ishikawa. Multi-label fashion image classification with minimal human supervision. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2261–2267, 2017. 1

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2014. 7

[16] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *International Conference on Machine Learning Deep Learning 2015 Workshop*, 2015. 5

[17] Piotr Koniusz, Yusuf Tas, and Fatih Porikli. Domain adaptation by mixture of alignments of second-or higher-order scatter tensors. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2, 2017. 1

[18] Piotr Koniusz, Yusuf Tas, Hongguang Zhang, Mehrtash Harandi, Fatih Porikli, and Rui Zhang. Museum exhibit identification challenge for the supervised domain adaptation and beyond. In *The European Conference on Computer Vision*, 2018. 1, 2, 6

[19] Piotr Koniusz and Hongguang Zhang. Power normalizations in fine-grained image, few-shot image and graph classification. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*. IEEE, 2020. 1

[20] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. 1

[21] Yuncheng Li, Yale Song, and Jiebo Luo. Improving pairwise ranking for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3617–3625, 2017. 6

[22] 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 *The European Conference on computer vision*, 2014. 2, 6

[23] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sungju Hwang, and Yi Yang. Learning to propagate labels: transductive propagation network for few-shot learning. In *International Conference on Learning Representations*, 2019. 3, 6

[24] François Pachet and Pierre Roy. Improving multilabel analysis of music titles: A large-scale validation of the correction approach. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(2):335–343, 2009. 1

[25] Guo-Jun Qi, Xian-Sheng Hua, Yong Rui, Jinhui Tang, Tao Mei, and Hong-Jiang Zhang. Correlative multi-label video annotation. In *Proceedings of the ACM international conference on Multimedia*, pages 17–26, 2007. 1

[26] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. *Proceedings of the international conference on machine learning*, 2015. 1

[27] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850, 2016. 1

[28] Fatemeh Shiri, Xin Yu, Fatih Porikli, Richard Hartley, and Piotr Koniusz. Identity-preserving face recovery from stylized portraits. *International Journal of Computer Vision*, 127(6-7):863–883, 2019. 1

[29] Fatemeh Shiri, Xin Yu, Fatih Porikli, Richard Hartley, and Piotr Koniusz. Recovering faces from portraits with auxiliary facial attributes. In *Winter Conference on Applications of Computer Vision*, pages 406–415, 2019. 1

[30] Christian Simon, Piotr Koniusz, Richard Nock, and Mehrtash Harandi. Adaptive subspaces for few-shot learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1

[31] Christian Simon, Piotr Koniusz, Richard Nock, and Mehrtash Harandi. On modulating the gradient for meta-learning. In *The European Conference on Computer Vision*, 2020. 1, 6

[32] Jake Snell, Kevin Swersky, and Zemel Richard. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 2017. 1, 3, 5

[33] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018. 1, 3, 5, 6, 8

[34] Y. Tas and P. Koniusz. Cnn-based action recognition and supervised domain adaptation on 3d body skeletons via kernel feature maps. *British Machine Vision Conference*, 2018. 1

[35] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, 2016. 1, 2, 5

[36] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. Cnn-rnn: A unified framework for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2285–2294, 2016. 3, 6

[37] Zhouxia Wang, Tianshui Chen, Guanbin Li, Ruijia Xu, and Liang Lin. Multi-label image recognition by recurrently discovering attentional regions. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 3, 6

[38] Jason Weston, Samy Bengio, and Nicolas Usunier. Wsabie: Scaling up to large vocabulary image annotation. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011. 6

[39] Hao Yang, Joey Tianyi Zhou, Yu Zhang, Bin-Bin Gao, Jianxin Wu, and Jianfei Cai. Exploit bounding box annotations for multi-label object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 280–288, 2016. 3

[40] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. DeepEMD: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[41] Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. CANet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[42] Hongguang Zhang and Piotr Koniusz. Model selection for generalized zero-shot learning. *The European Conference on Computer Vision (TASK-CV workshop)*, 2018. 1

[43] Hongguang Zhang and Piotr Koniusz. Zero-shot kernel learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7670–7679, 2018. 1

[44] Hongguang Zhang and Piotr Koniusz. Power normalizing second-order similarity network for few-shot learning. *Winter Conference on Applications of Computer Vision*, 2019. 1

[45] Hongguang Zhang, Piotr Koniusz, Songlei Jian, Hongdong Li, and Philip H. S. Torr. Rethinking class relations: Absolute-relative supervised and unsupervised few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9432–9441, 2021. 1

[46] Hongguang Zhang, Jing Zhang, and Piotr Koniusz. Few-shot learning via saliency-guided hallucination of samples. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1

[47] Hongguang Zhang, Li Zhang, Xiaojuan Qi, Hongdong Li, Philip H. S. Torr, and Piotr Koniusz. Few-shot action recognition with permutation-invariant attention. In *The European Conference on Computer Vision*, 2020. 1

[48] Manli Zhang, Jianhong Zhang, Zhiwu Lu, Tao Xiang, Mingyu Ding, and Songfang Huang. IEPT: Instance-level and episode-level pretext tasks for few-shot learning. In *International Conference on Learning Representations*, 2021. 1

[49] Ruixiang ZHANG, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. MetaGAN: An adversarial approach to few-shot learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 1

[50] Rui Zhang, Yusuf Tas, and Piotr Koniusz. Artwork identification from wearable camera images for enhancing experience of museum audiences. In *Museums and the Web*, 2017. 1

[51] Shan Zhang, Dawei Luo, Lei Wang, and Piotr Koniusz. Few-shot object detection by second-order pooling. *Asian Conference on Computer Vision*, 2020. 1

[52] Dengyong Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in neural information processing systems*, pages 321–328, 2004. 4

[53] Feng Zhu, Hongsheng Li, Wanli Ouyang, Nenghai Yu, and Xiaogang Wang. Learning spatial regularization with image-level supervisions for multi-label image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5513–5522, 2017. 3, 6

[54] Hao Zhu and Piotr Koniusz. REFINE: Random RangE FInder for Network Embedding. In *ACM International Conference on Information and Knowledge Management*, 2021. 1

[55] Hao Zhu and Piotr Koniusz. Simple spectral graph convolution. In *International Conference on Learning Representations*, 2021. 1

[56] Hao Zhu, Graham Neubig, and Yonatan Bisk. Few-shot language coordination by modeling theory of mind. In *International Conference on Machine Learning*, 2021. 1

[57] Hao Zhu, Ke Sun, and Piotr Koniusz. Contrastive laplacian eigenmaps. In *Conference on Neural Information Processing Systems*, 2021. 1