Few-shot Partial Multi-label Learning with Synthetic Features Network

Yifan Sun
Shandong University
Yunfeng Zhao
Shandong University
Guoxian Yu (gxyu@sdu.edu.cn)
Shandong University
Zhongmin Yan
Shandong University
Carlotta Domeniconi
George Mason University

Research Article

Keywords: Partial Multi-label Learning, Few-shot Learning, Weakly-supervised Learning, Noisy Labels, Label Correlations, Data augmentation

Posted Date: January 2nd, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2421677/v1

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Knowledge and Information Systems on September 28th, 2023. See the published version at https://doi.org/10.1007/s10115-023-01988-2.

Few-shot Partial Multi-label Learning with Synthetic Features Network

Yifan Sun[1,2], Yunfeng Zhao[1,2], Guoxian Yu[1,2,*], Zhongmin Yan[1,2], Carlotta Domeniconi[3]
[1]School of Software, Shandong University, Jinan, China
[2]Joint SDU-NTU Centre for Artificial Intelligence Research (C-FAIR), Shandong University, Jinan, China
[3]Department of Computer Science, George Mason University, VA, USA
Email: {yfsun, yunfengzhao}@mail.sdu.edu.cn; {gxyu, yzm}@sdu.edu.cn, carlotta@cs.gmu.edu
Corresponding author: Guoxian Yu.

Abstract

In partial multi-label learning (PML) problems, each training sample is partially annotated with a candidate label set, among which only a subset of labels are valid. The major hardship for PML is that its training procedure is prone to be misled by false positive labels concealed in the candidate label set. To train a noise-robust multi-label predictor for PML problem, most existing methods hold the assumption that sufficient training samples are available. However, in actual fact, especially when dealing with new tasks, we more often only have a few PML samples for the target task. In this paper, we propose a unified model called FsPML-SF (Few-shot Partial Multi-Label Learning with Synthetic Features Network). FsPML-SF includes three modules: label disambiguation, data augmentation and classifier induction. Specifically, FsPML-SF attempts to update the label credibility of each PML sample by leveraging the feature and semantic similarities, the label credibility of other samples and label co-occurrence in a unified objective function. Next, FsPML-SF introduces a synthetic feature network to generate more training samples from pairs of given samples with corresponding label credibility values. FsPML-SF then utilizes the original and synthesized samples to induce a noise-tolerant multi-label classifier. We conducted extensive experiments on benchmark datasets, FsPML-SF outperforms recent competitive PML baselines and few-shot solutions. Both the label denoising and data augmentation improve the performance of PML on few-shot data.

Keywords: Partial Multi-label Learning, Few-shot Learning, Weakly-supervised Learning, Noisy Labels, Label Correlations, Data augmentation

1. Introduction

As a novel weakly supervised learning framework, Partial Multi-label Learning (PML) [1, 2] models the scenario where each instance is associated with a set of candidate labels, but only a subset of these labels corresponds to the ground-truths. Because of the difficulty

and expense to obtain precise annotations, PML is more practical in real-world applications comparing with the prevalent classification problem multi-label learning (MLL) [3], where each sample is assigned with multiple valid labels simultaneously. Specifically, the task of PML naturally arises in crowdsource annotations. In such a scenario, the object might be annotated with multiple labels provided by different annotators to form the candidate label set, which is usually overcomplete and contains irrelevant labels [4]. Recent years have witnessed an growing research and application of PML in various domains, such as image analysis [5, 6], text mining, gene function prediction [2, 7], and so on.

PML aims to train a classifier from partially labeled data so as to predict the correct labels for an unseen instance automatically. The key challenge of PML is how to deal with the ambiguities caused by the irrelevant labels in candidate label set. One straightforward way is to simply treat all candidate labels as valid ones, and then adapt any off-the-shelf multi-label classification method to induce the classifier. However, such strategy ignores the false positive labels concealed in the candidate label set, which would significantly mislead the learning process and degrade the performance of learning model. To tackle this problem, many approaches follow the label disambiguation strategy to elicit the ground-truth labels from candidate label set and then adopt the elicited labels to induce the classifier. They usually define a confidence score to predict the probability for each candidate label to be ground-truth one. For example, the smooth assumption that similar (dissimilar) samples have similar (dissimilar) label assignments is utilized to extract high confidence labels [8, 9]. While others focus on the sparsity constraint of the latent ground-truth label matrix [2, 10, 11]. In addition, the label correlation is also employed to recover label confidence [1, 12]. A recent work [13] proposed a new perspective for PML that considers the label information is precise while the feature information is missing, and re-interpret the task of PML as a Feature Completion problem.

Although these solutions have shown improvement of practical performance for solving PML tasks, most of them hold an implicit/explicit assumption that sufficient data is available to induce the classifier. But in practice, acquiring sufficiently annotated/training samples is an expensive and even infeasible task, which consume huge manual power and financial resources. Thus, in real-world scenarios, it's more practical to perform partial multi-label classification with only few-shot data. In such scenario, these methods will suffer from the data limitation and fail to perform well, as shown in our experiments. Although Xie *et. al.* [14] recently proposed a solution named PML-MD [14], which performs disambiguation in a meta-learning fashion, it still requires abundant samples for inducing the classifier. In addition, Few-shot Multi-Label Learning (FsMLL) [15, 16] methods and zero-shot multi-label learning approaches [17, 18] are also incapable to tackle this problem. The irrelevant labels concealed in the candidate label set will seriously mislead these methods when generalizing to the target task, causing a compromised performance. Due to the difficulty to obtain extensive and precisely annotated samples in most real-world scenarios, partial multi-label learning on few-shot data is a task with practical significance but under-studied yet. To tackle this problem, a recent proposed work FsPML [19] aims to rectify the positive and negative prototypes of labels in the prototype network framework [20]. Despite the advances FsPML has achieved, a potential limitation is its representation ability of the prototypes. When

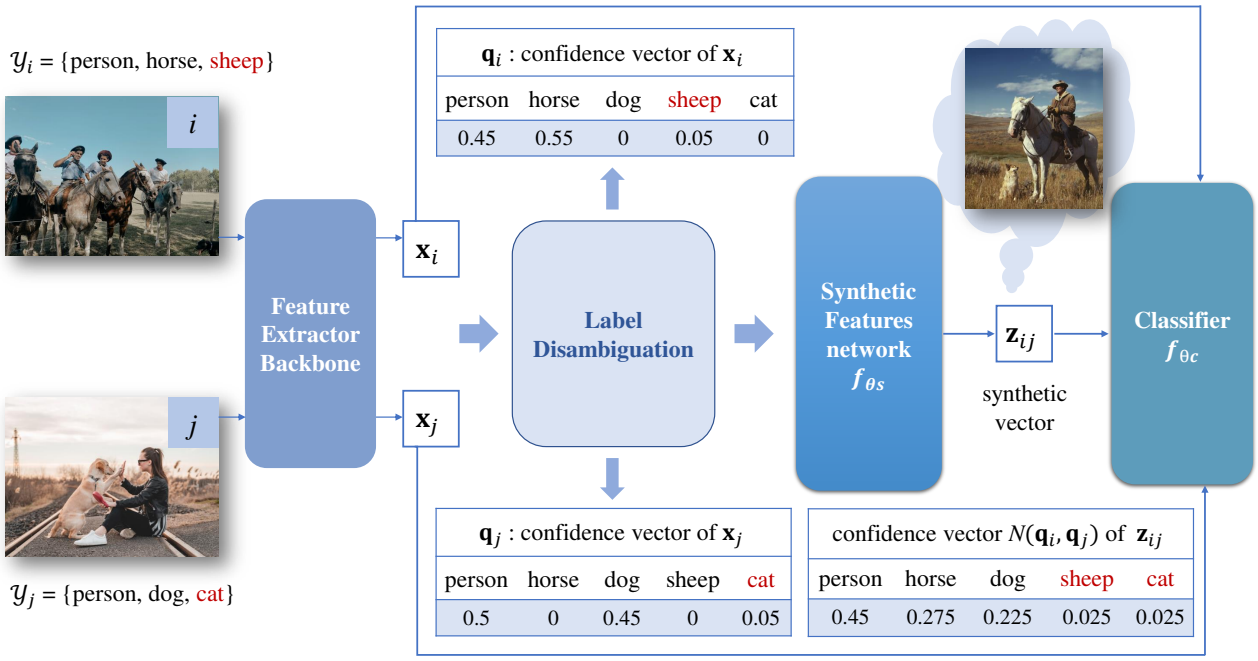


Figure 1: The overall schematic framework of FsPML-SF. Given a pair of samples and their candidate labels (the red labels are *irrelevant* ones), FsPML-SF first performs label disambiguation by leveraging the feature and label similarity between samples, label credibility of other samples and label correlations to update the label confidence vector for each sample. Next, we send the given pairs of samples with their corresponding label confidence vectors into the synthetic features network f_{θ_s}. The synthetic features network is designed to learn label-specific features and to synthesize new samples with credible labels. Both the original and generated samples are then used to induce the multi-label classifier f_{θ_c}.

applying to new tasks, due to the small number of relevant samples per label, the prototypes may be biased towards several labels [21]. Besides, FsPML predicts the ground-truth label by measuring the distance between embedded sample and the label's positive and negative prototypes. It simplifies the multi-label classification problem into a binary one, thus fails to explicitly model the important correlations between all candidate labels. To sum up, existing PML solutions are restricted to the quantity of training samples, and FsMLL methods are incapable to handle irrelevant annotations, while FsPML is still limited by the representation ability of the prototypes and fails to leverage PML data in a sensible way. Therefore, how to implement partial multi-label learning in few-shot scenarios is still an under-studied problem.

This paper studies an under-studied and challenging few-shot multi-label classification using scarce samples with noisy labels. For this purpose, we propose a unified method called FsPML-SF (Few-Shot Partial Multi-Label Learning with Synthetic Features Network). The FsPML-SF model comprises three component: *label disambiguation*, *data augmentation* and *classifier induction*. In the label disambiguation procedure, FsPML-SF disambiguates the label confidence vector for each PML sample in a rational way. The feature and semantic similarities, the label credibility of other samples and label co-occurrence are collaboratively utilizing into a unified objective function. In the data augmentation procedure, given a pair of

samples with their confidence vectors, FsPML-SF introduces a synthetic features network to generate a new feature vector with corresponding label credibility values, which also encode the label correlations of training data. In the classifier induction procedure, the original and synthesized samples are both utilized to induce a noise-tolerant multi-label classifier. Note that these three procedures are operated in a reciprocal reinforcement manner by a unified framework, and we develop an alternative optimization strategy to optimize them. The whole framework of FsPML-SF is illustrated in Figure 1.

This paper is a major extension of our previous work [22]. As new material, this paper contains an extended discussion on few-shot multi-label classification and a more in-depth analysis of the experimental results. In addition, we conduct various simulated experiments to achieve better understanding of our model. Our main contributions are summarized as follows:

(i) We focus on a typical and practical few-shot multi-label learning problem and propose an approach (FsPML-SF) to achieve FsPML from a new perspective, allowing the generation of new multi-label samples with credible labels by pairing limited training samples. The label-informative features with high credibility are highlighted in the synthetic vectors. Therefore, FsPML-SF surmounts the bottleneck of scarce training samples for inducing the noise-resistant multi-label classifier.

(ii) In the label disambiguation stage, FsPML-SF dislodges irrelevant labels of training data in a sensible way by leveraging the feature and semantic similarity between pairwise samples, label co-occurrence, credible labels of other samples.

(iii) We conduct extensive experiments on benchmark multi-label datasets to demonstrate that FsPML-SF significantly outperforms the related and competitive PML methods [1, 2, 8, 23, 13], and few-shot MLL methods [15, 16]. Both the label denoising and data augmentation improve the performance of PML on few-shot data.

The paper is organized as follows. Section 2 reviews related work in the fields of partial multi-label learning and few-shot multi-label learning. Section 3 explains the technical details of our proposed approach. Section 4 reports the experimental results and analysis, and Section 5 presents our conclusions and suggestions for future work.

2. Related Work

Few-shot partial multi-label learning is closely relevant to two popular learning frameworks: partial multi-label learning [1, 2] and few-shot multi-label learning [15, 16].

2.1. Partial Multi-label Learning

Partial multi-label learning (PML) is a new and more challenging branch of the standard multi-label learning (MLL) [3], where each sample is assigned with into multiple classes simultaneously. Besides, PML also differs from the popular partial-label learning (PLL), which assumes only one label from candidate set of the sample is valid [24, 25]. With the annotations provided by multiple annotators under the crowdsourcing setting, the PML problem arises naturally in real world scenarios, where each training sample is not only tagged

with the ground truth labels, but also with some irrelevant ones [26]. The following gives a brief review of popular PML solutions.

PML with feature prototype/label correlation (PML-fp/PML-lc) [1] assigns a confidence value for each candidate label, and then optimizes the confidence values by further exploiting feature prototypes or label correlations. Feature-induced PML solution (fPML) [2] jointly factorizes the observed sample-label association matrix and the sample-feature data matrix into low-rank ones to identify irrelevant labels and optimizes a multi-label predictor with respect to low-rank label matrix. PML-LFC [27] considers the negative correlations between features and labels and estimates the confidence values of relevant labels for each instance using both feature and semantic similarities. PML-NI [23] identifies the noisy labels under the observation that noisy labels are caused by some ambiguous features of the sample. MUSER [28] jointly considers redundant labels together with noisy features during the training process using feature similarity and label correlation. PML-LCom [29] utilizes label compression to improve the performance and efficiency of PML on datasets with large label spaces. PML-LCom first splits the observed label matrix into a latent relevant label matrix and a noisy one. Next, it coordinates relevant label matrix learning using the feature data matrix, and trains a multi-label predictor with respect to the compressed label matrix. HALE [30] formulates the task of PML as a instance-to-label matching selection problem and introduces a graph matching algorithm with many-to-many constraint to accommodate to the PML problem. SSPML [31] tackles the PML problem in semi-supervised setting and uses a latent label variable for each example as the low-dimensional embedding of the feature space. The multi-label classifier is jointly trained under the supervision of label variables. Sun *et al.* [32] attempted to simultaneously remove noisy outliers from the training instances and train robust partial multi-label classifier for unlabeled instances prediction. FIMAN [33] tackles the multi-view PML problem. The affinity information conveyed by different views are adaptively fused to disambiguate candidate label set by enforcing manifold structure preservation in the label space. MILI-PML [34] is derived from a clear probabilistic formulation and it naturally incorporates the feature/label relevancy considerations. Sun *et al.* [35] proposed a Global-Local Label Correlation approach for PML. The global structure information of labels is explicitly exploited via a label coefficient matrix and the local label correlations are captured with a new label manifold regularizer. PML-SALC [36] presents PML based on sparse asymmetric label correlations, which utilizes the sparse asymmetric label correlation matrix to alleviate the negative influence of noisy labels to obtain label confidence. PML-LMNNE [37] conducts disambiguation by projecting labels and features into a lower-dimension embedding space and reorganizes the underlying structure by LMNN in the embedding space simultaneously. Besides relying on extra assumptions on the data structures, PML-GAN [38] defines a disambiguation network to identify irrelevant labels and induces a multi-label predictor to map the training samples to their disambiguated label vectors. PML-MT [39] iteratively refines the label confidence matrix through a couple of selfensemble teacher networks and trains two prediction networks simultaneously in a mutual teaching manner. PML-MD [14] tries to disambiguate in a meta-learning fashion. The multi-label classifier is trained by minimizing a confidence-weighted ranking loss while the confidence for each candidate label is adaptively estimated by its performance on a small validation set. A recent work NATAL [13] assumes

the labeling information is precise while the feature information is partially corrupted. NATAL models the PML task as a feature completion problem, and induces the prediction model from completed features using candidate labels.

There are also some methods follow a two-stage strategy [8, 9, 12] that firstly attempt to obtain credible labels and then take the elicited labels to induce a multi-label classifier. To name a few, PARTICLE [8] firstly estimates the confidence of candidate label for each PML training example via iterative label propagation, and then induces a multi-label predictor using credible labels with high label confidence. DRAMA [9] firstly optimizes the confidence value for each label by the feature manifold, and then induces a gradient boosting model to fit the learnt label confidences. PML-LD [12] recovers the label distribution by the topological information of feature space and label correlations, and then trains a multi-label predictive model by fitting a regularized multi-output regressor with the recovered label distributions.

These aforementioned PML solutions assume that a large amount of training samples are available to train the predictor, and they are incapable to deal with a new task with limited samples. Recently, Zhao *et al.* [19] proposed an approach called FsPML to address PML in the few-shot scenario. FsPML learns an embedding network to rectify the positive and negative prototypes of each label. An unseen sample can then be classified via its distance to positive and negative prototypes of each label. However, the representation capability of prototypes restricts its performance, and it disregards the label correlations and the relevance of different neighbourhood samples. In comparison, FsPML-SF updates the credibility scores of samples using both the label and feature similarity values, as well as the label co-occurrence. It introduces a synthetic feature network to generate new samples with label confidence values utilizing pairing samples. Thus, FsPML-SF turns the few-shot problem into a many-shot one by data augmentation for inducing the noise-robust multi-label classifier.

2.2. Few-shot Multi-label Learning

Few-shot multi-label learning (FsMLL) aims to learn a multi-label predictor based on a handful samples for the target task, and it has been recently explored in many areas. For example, ZAGCNN [40] is a few- and zero-shot methods for multi-label text classification by matching discharge summaries in electronic medical records using feature vectors. Alfassy *et al.* [15] leveraged LaSO (Label Set Operations networks) to manipulate the 'semantic content' of the samples in feature space and produce samples containing the intersection, union or set-difference of labels present in input samples and sythesize new samples for multi-label few-shot classification. Hou *et al.* [41] studied the few-shot multi-label classification for user intent detection. They firstly learnt universal thresholding experience on data-rich domains, and then adapted the thresholds to certain few-shot domains with a calibration based on non-parametric learning. Simon *et al.* [42] aimed to extend some off-the-shelf few-shot single-label learning solutions to work in the multi-label regime and introduced a neural module to estimate the label count of a given sample by exploiting the relational inference. DESIRENet [43] maps the features into the semantic embedding space via label word vectors to exploit the label correlation and introduces a novel semantic inference mechanism for leveraging prior knowledge learned from historical labels. LARN [44] tackles the problem of semi-supervised few-shot multi-label node classification by taking advantage of the semantic knowledge of la-

bels to characterize nodes and their neighbors. A label correlation scanner is then proposed to adaptively capture the label correlation and extract the useful information to generate the final node representation. KGGR [16] is a knowledge-guided graph routing framework, it unifies prior knowledge of statistical label correlations with deep neural networks for the target novel task.

These FsMLL methods [15, 16, 40, 41, 42, 43, 44] adopt the ideal assumption that the samples are annotated with precise labels. However, in actual fact, it takes more energy and expense to meet this precise annotations premise. It is more common that a set of candidate labels are roughly assigned by annotators, on which FsMLL methods fail to perform well and suffer a greatly compromised performance. Our FsPML-SF aims to achieve FsMLL in a more robust setting. Different from LaSO, which generates feature vectors by manipulating a pair of label sets, FsPML-SF firstly extracts each sample's label-informative features and recombine them utilizing their label confidence values to synthesize new samples. Thus, the synthetic vectors can resist with the negative impact of noisy labels, and FsPML-SF outperforms those FsMLL solutions.

3. Proposed Method

3.1. Problem Formulation and Notation

Let $\mathcal{X} \in \mathbb{R}^d$ denote the d-dimensional feature space, and $\mathcal{Y} \in \{0|1\}_{c=1}^{m}$ be the label space with m distinct labels. Given an N-way K-shot training dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)|1 \leq i \leq n\}$, where N stands for the number of labels, K is the number of samples tagged with each label, $\mathbf{x}_i \in \mathcal{X}$ is the feature vector of the i-th sample, and $\mathbf{y}_i$ is the multi-hot label vector of $\mathbf{x}_i$, which encodes the set of candidate labels $\mathcal{Y}_i \subset \mathcal{Y}$ annotated to $\mathbf{x}_i$. PML holds the assumption that the ground-truth labels $\tilde{\mathcal{Y}}_i \subset \mathcal{Y}_i$. Thus, $\tilde{\mathcal{Y}}_i$ cannot be directly used for inducing the predictor. The task of FsPML-SF is to train a multi-label classifier $f_{\theta_c} : \mathcal{X} \to 2^{\mathcal{Y}}$ from $\mathcal{D}$, which can precisely predict the ground-truth label set of an unseen sample. Table 1 summarizes the frequently-used symbols.

3.2. Few-shot Partial Multi-label Learning with Synthetic Features Network

To handle the lack of ground-truth labels of training samples, we let $\mathbf{Q} = [\mathbf{q}_1, \cdots, \mathbf{q}_n]^T$ be the latent label confidence matrix, where q_{ik} denotes the confidence value of the k-th label being the ground-truth of $\mathbf{x}_i$. Different from existing two-stage approaches [8, 9, 12] that firstly obtain the credible labels and then utilize these credible labels to train the multi-label classification model, FsPML-SF operates the three procedures in a unified framework. We perform the label disambiguation by updating $\mathbf{Q}$, generate new samples by the synthetic feature network f_{θ_s} and induce the multi-label predictor f_{θ_c} in a reciprocal reinforcement manner, which we will discuss in the following subsections.

3.2.1. Label Disambiguation

In this phase, FsPML-SF targets to eliciting the label credibility values for each sample via collaboratively leveraging the label and feature similarity values, label co-occurrence and labels of other samples. Many existing PML methods operate in the feature space based on

Table 1: Notation table

Notation	Description
Networks	
f_{θ_s}	The synthetic features network
f_{θ_c}	The multi-label classifier
Indices and Trade-off parameters	
n	The number of samples within $\mathcal{D}$
m	The number of labels within $\mathcal{D}$
n_g	The number of generated samples
n_{kh}	The number of samples whose candidate set includes label k and h
$*^{(\tau)}$	The * in τ-th iteration for alternative optimization
$\alpha/\beta/\lambda$	The trade-off parameters
Parameters in the label disambiguation stage	
$\mathbf{x}_i$	The feature vector of the i-th sample
$\mathbf{y}_i$	The label vector of the i-th sample
$\mathbf{Q} = [\mathbf{q}_1, \cdots, \mathbf{q}_n]^T$	The label confidence matrix of samples
$\mathbf{S} = [s_{ij}]_{n\times n}$	The feature similarity matrix
$\mathbf{T} = [t_{ij}]_{n\times n}$	The semantic similarity matrix
$\mathbf{P} = [p_{kh}]_{m\times m}$	The label co-occurrences statistics matrix
Parameters in the data augmentation stage	
$\mathcal{X}_i$	The feature map of the i-th sample
$\mathcal{C} = [\mathbf{C}_1, \mathbf{C}_2, \cdots, \mathbf{C}_m]$	The label-specific activation maps in f_{θ_s}
$[\mathbf{l}_1^i, \mathbf{l}_2^i, \cdots, \mathbf{l}_m^i]$	The content-aware label-specific features of the i-th sample
$\tilde{\mathbf{x}}_i$	The label-confidence aware vector of the i-th sample
$\mathbf{z}_{ij}$	The synthetic vector generated from the i-th and the j-th sample

the smooth assumption that similar samples have similar label assignments, either to elicit credible labels [8] or perform label enhancement [12]. However, most of them only utilize neighborhood samples while ignore the less similar ones. Here, limited by the number of training samples, we need to use the PML samples in a more rational way, instead of only considering neighborhood ones.

To make a better label disambiguation, FsPML-SF updates the label confidence value by calculating the similarity between samples from both the feature and label space as well as referring to other sample-label credibility. In addition, some labels tend to more often co-annotate to the same samples. We design a label co-occurrence statistics matrix $\mathbf{P}$ based on the co-occurrence patterns. For this purpose, we update $\mathbf{Q}$ by minimizing the disambiguation

loss as:

$$\begin{aligned}\Omega_1(\mathbf{Q}) = &\sum_{\mathbf{x}_i\in\mathcal{D}}\sum_{k\in\mathcal{Y}_i}(q_{ik} - \sum_{\mathbf{x}_j\in\mathcal{D},j\neq i}[\![q_{jk}\neq 0]\!]s_{ij}t_{ij}q_{jk})^2 \\ &+\lambda\sum_{\mathbf{x}_i\in\mathcal{D}}\sum_{k\in\mathcal{Y}_i}\sum_{h\in\mathcal{Y}_i,h\neq k}[\![q_{ik}q_{ih}\neq 0]\!](q_{ik}q_{ih}-p_{kh})^2 \\ &s.t.\ \mathbf{Q}\geq 0,\ \sum_{k\in\mathcal{Y}_i}q_{ik}=1\end{aligned} \tag{1}$$

where s_{ij} is the feature similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$, t_{ij} is the semantic similarity derived from label confidence vectors $\mathbf{q}_i$ and $\mathbf{q}_j$, p_{kh} is the label co-occurrence statistics of label k and h, λ is the trade-off parameter. $[\![q_{jk}\neq 0]\!]$ ($[\![q_{ik}q_{ih}\neq 0]\!]$) returns 1 if the condition holds, and 0 otherwise. The first constraint guarantees that each candidate label has a non-negative confidence value, and the second restricts the confidence value is within $[0,1]$, and the sum of them equals to 1.

The first term of $\Omega_1(\mathbf{Q})$ collaboratively takes the feature similarity, semantic similarity and sample-label credibility into account. When disambiguating a specific label (*e.g.* label k) of sample i, we generally refer to all the other samples in $\mathcal{D}$ which also annotated with label k to compute the label confidence. For simplicity, we define the to-be disambiguated $\mathbf{x}_i$ as a *needy* sample, while the other samples annotated with the same label k as *assistant* samples. The label-credibility of the *assistant* samples can provide some reference helpful for the label disambiguation of q_{ik}. While different *assistant* samples have different relevance for updating the label confidence of $\mathbf{x}_i$. If an *assistant* sample is more similar with $\mathbf{x}_i$ in feature and label space, its label-confidence should exert greater influence on disambiguating $\mathbf{x}_i$. To evaluate the influence of an *assistant* sample towards disambiguating the label of $\mathbf{x}_i$, we compute the product of feature and semantic similarity (s_{ij} and t_{ij}) between them as the weight to quantify this effect. The first term of $\Omega_1(\mathbf{Q})$ commendably accounts for different cases, as discussed below and illustrated in Fig. 2.

Case a: If $\mathbf{x}_j$ is highly similar with $\mathbf{x}_i$ in both feature and label space (high values of s_{ij} and t_{ij}), then label $k\in\mathcal{Y}_i$ shared by $\mathbf{x}_j$ is more credible for $\mathbf{x}_i$, which means a high confidence value q_{ik} and label k is a highly probable ground-truth for $\mathbf{x}_i$.

Case b: If $\mathbf{x}_j$ has high feature and semantic similarities with $\mathbf{x}_i$, but a low q_{jk}, which means that label k is less related with $\mathbf{x}_j$. This also drags down q_{ik} and dislodges label k from $\mathcal{Y}_i$.

Case c: If $\mathbf{x}_j$ is dissimilar with $\mathbf{x}_i$ in both the feature and label space with a low q_{jk}, then $\mathbf{x}_j$ provides little information for disambiguating label k of $\mathbf{x}_i$. Note that FsPML-SF pairs all other assistant samples with $\mathbf{x}_i$, the dissimilar samples has tiny impact on the overall disambiguation effect on $\mathbf{x}_i$.

Case d: If the product of two similarities is low but q_{jk} is high, it is less likely for $\mathbf{x}_i$ to tag label k. In other words, $\mathbf{x}_j$ has little influence on q_{ik}.

It's worth to note that that these above four cases are discussed just for better understanding the model of the first term of $\Omega_1(\mathbf{Q})$. We want to showcase that different *assistant* samples help the disambiguation of *needy* sample with different effect. We do not explicitly distinguish the "high" and "low" in semantic and feature similarities in our function. Be-

sides, if the product of feature and semantic similarity is moderate, it means this pair have an intermediate influence towards the disambiguation.

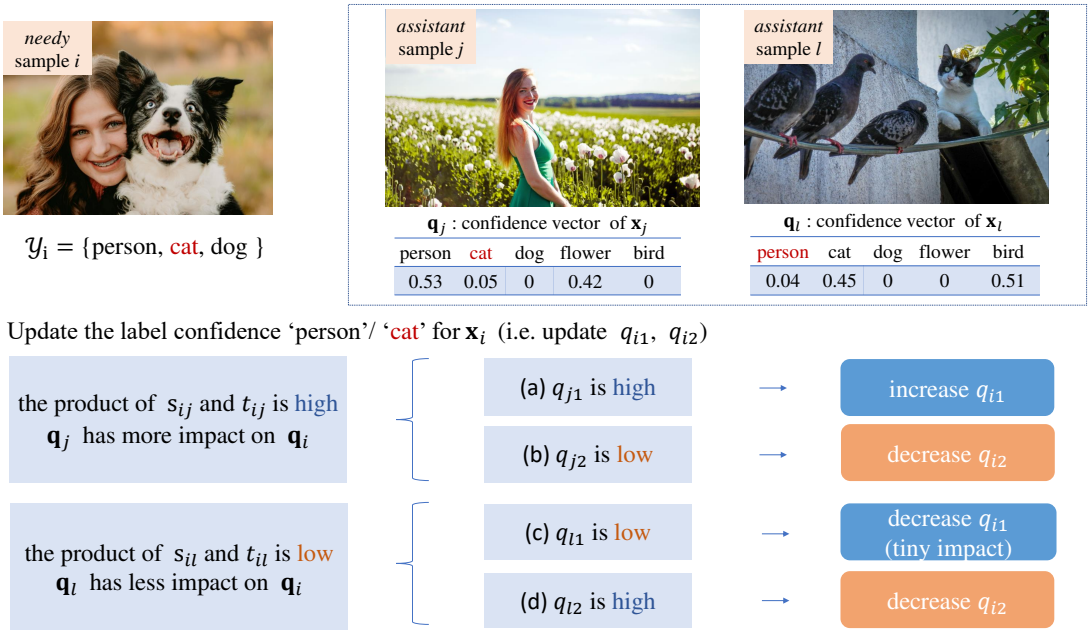


Figure 2: A toy example for illustrating the first term of $\Omega_1(\mathbf{Q})$. When disambiguating the labels of $\mathbf{x}_i$, FsPML-SF generally considers the feature and semantic similarities (s_{ij} and t_{ij}) with other samples as well as their label confidences (red labels are *irrelevant* ones). In the above cases, FsPML-SF tends to raise the confidence value of 'person' and decrease the value of 'cat' for $\mathbf{x}_i$.

The second term targets at accounting for the label correlation of all training samples. It's set up follow the observation that some labels (such as 'ocean' and 'ship') usually have high co-occurrence frequency, while other pairs (*e.g.* 'sunny' and 'fog') may never annotate together to the same sample. Therefore, it's significant to incorporate the label correlations for better disambiguation. For this purpose, we define a label correlation matrix $\mathbf{P} = [p_{kh}]_{m\times m}$, where p_{kh} denotes the co-occurrence statistics of label k and h. For every label pair k and h of one sample, $\Omega_1(\mathbf{Q})$ encourages the product of their label confidence values as close to p_{kh} as possible. Thus effectively increases the confidence values of frequently co-occurring label pairs and reduces the credibility scores of rarely co-annotated label pairs. Specifically, we first setup $\mathbf{P}$ by the number of samples whose candidate label set contains both label k and h and then normalize it by dividing the number of samples with k or h. In the follow-up iterations, $\mathbf{P}$ is updated based on the latest label confidence matrix $\mathbf{Q}$, thus the negative impact of noisy labels is greedily diminished, while the label co-occurrence (correlation) is explored and exploited to disambiguate labels. In this way, $\Omega_1(\mathbf{Q})$ can obtain credible label confidence matrix $\mathbf{Q}$ by mining PML samples in a sensible way. The incorporation and computation of label co-occurrence matrix $\mathbf{P}$ will be mentioned in detail in the optimization section.

3.2.2. Data Augmentation

Different from usual PML problem setting, which allows multiple samples utilized to train a noise-robust multi-label predictor, in FsPML scenario, we only have a handful of few-shot training PML samples. Despite $\Omega_1(\mathbf{Q})$ attempts to use all the PML samples to disambiguate the labels of the target sample, we still expects more PML samples to better induce the classifier. The existing FsPML [19] borrows the idea of meta learning in a prototype network manner to learn the prototypes of each label and disambiguate the labels in the prototype space, but it neglects the label correlations, and maybe biased toward several labels in the target task. Given two training samples $\mathbf{x}_i$ and $\mathbf{x}_j$, and their label confidence vectors $\mathbf{q}_i$ and $\mathbf{q}_j$, we introduce the synthetic features network f_{θ_s} to generate a new feature vector whose corresponding soft labels are made up of $\mathbf{q}_i$ and $\mathbf{q}_j$. Then we can incorporate the generated samples along with the soft labels to augment the original few-shot samples and to induce the multi-label predictor. This is motivated by the intuition that if a human observe a hypothetical image synthesised from $\mathbf{x}_i$ and $\mathbf{x}_j$, he/she will more likely to attain semantics labels in $\mathbf{q}_i$ and $\mathbf{q}_j$ to this generated samples. For example, given two animal images, the synthesised image should be more relevant to the high confidence label 'zoo' shared by these two samples. The generated samples highlight the label-informative features with high confidence values, which is more conducive to the training of classification networks.

There also exist some prior works to generate new examples based on the available few-shot ones [45, 46, 47]. Simple image transformations (horizontal flips, scaling, shifts), have been exploited from the beginning. In other works, some methods perform example synthesis using additional semantic information[16]. Specifically, a strong recent trend is to generate examples using Generative Adversarial Networks (GANs) [48], while these methods are prone to mode collapse due to the limited training samples quantity. PML-GAN operates in the GAN framework, but it disambiguates noisy labels in an adversarial learning manner, instead of generating new samples for inducing multi-label classifier. LaSO [15] aims to generate new multi-label samples by combining few-shot samples and thus turns the few-shot multi-label learning problem into a many-shot one. However, LaSO performs data augmentation with certain label set operations and precise annotations. Due to the irrelevant labels of few-shot PML samples, the samples generated by LaSO are with low quality and will seriously compromise its performance.

FsPML-SF proposes the idea of generating samples with corresponding label confidence scores. Thus, we should give different labels with different attentions considering the input samples' confidence scores. We follow the work of Semantic Attention Module (SAM) in ADD-GCN [49] and decouple different label contents of given feature vectors. Next, we recombine these features into a label-confidence aware representation according to their label confidence values. The architecture of synthetic features network f_{θ_s} is sketched in Fig3.

Specifically, in synthetic features network f_{θ_s}, we use feature maps $\mathcal{X}_i \in \mathbb{R}^{h\times w\times d}$ in the h-height, w-width and d-dimensional feature space without average pooling instead of taking InceptionV3 [50] pre-processed feature vectors $\mathbf{x}_i \in \mathbb{R}^d$ as the input to get better pattern position information. FsPML-SF computes a label-specific activation maps $\mathcal{C} = [\mathbf{C}_1, \mathbf{C}_2, \cdots, \mathbf{C}_m] \in \mathbb{R}^{h\times w\times m}$ to convert the i-th sample's feature map $\mathcal{X}_i$ into a set of content-aware label-specific features $[\mathbf{l}_1^i, \mathbf{l}_2^i, \cdots, \mathbf{l}_m^i] \in \mathbb{R}^{d\times m}$, each of which describes the

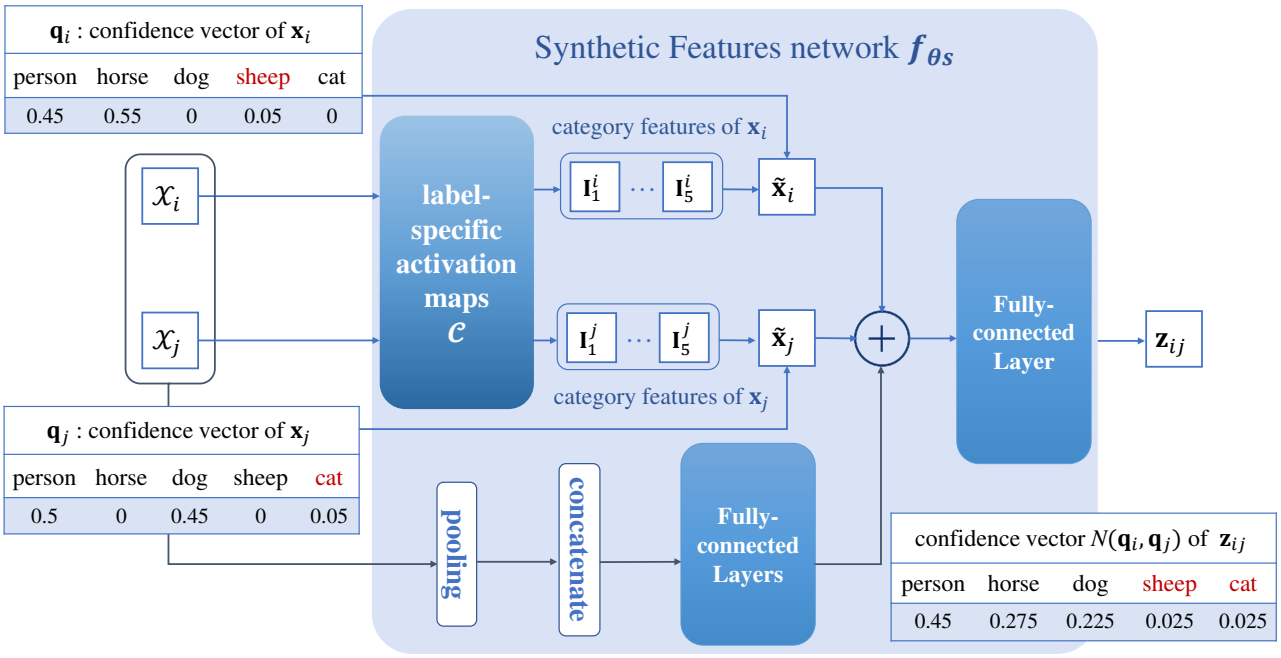


Figure 3: The architecture of synthetic features network f_{θ_s}.

contents related to a specific label from input features. Then, we perform Global Average Pooling (GAP) on the feature map and classify these pooled features with a two-dimensional convolution layer as the classifier. Next, the classifier is used to identify the label-specific activation maps by convolving the weights of classifier with feature map. Each label feature vector $\mathbf{l}_k^i$ is formulated as a weighted sum on $\mathcal{X}_i$ as follows:

$$\mathbf{l}_k^i = \mathbf{C}_k^T \mathcal{X}_i = \sum_{l=1}^{h} \sum_{r=1}^{w} \mathbf{C}_k^{(l,r)} \mathcal{X}_i^{(l,r)} \tag{2}$$

where $\mathbf{C}_k^{(l,r)}$ and $\mathcal{X}_i^{(l,r)}$ are the weight of k-th activation map $\mathbf{C}_k$ and the feature vector of the feature map $\mathcal{X}_i$ at (l, r), respectively. By doing so, $\mathbf{l}_k^i$ can selectively aggregate features related to label k. We want to remark that here we just showcase FsPML-SF on the typical image datasets, other non-image datasets can be also applied by replacing this convolution layer with other domain-specific networks (i.e., fully-connected layers for text datasets, or a three-dimensional convolution layer for audio datasets).

The learnt features are associated with different labels with varying confidence values, and we expect the synthetic feature vector with high label credibility to account for the principal parts, while features with low label confidence appear less. For this purpose, we define a label-confidence aware vector $\tilde{\mathbf{x}}_i \in \mathbb{R}^d$ as follows:

$$\tilde{\mathbf{x}}_i = \sum\nolimits_{k=1}^{m} q_{ik} \mathbf{l}_k^i \tag{3}$$

Here $\tilde{\mathbf{x}}_i$ is the transformed label-confidence aware representation of $\mathbf{x}_i$.

To generate label-confidence aware new samples, f_{θ_s} concatenates $\mathbf{x}_i$ and $\mathbf{x}_j$ together and send them into two blocks of fully-connected layer followed by batch-normalization, leaky-RELU, and dropout. Then, f_{θ_s} adds the output of two layers, $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_j$ together and send them into the last fully-connected layer followed by leaky-RELU. Thus, we get the synthetic feature vector $\mathbf{z}_{ij} = f_{\theta_s}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{q}_i, \mathbf{q}_j) \in \mathbb{R}^d$, here $f_{\theta_s}(\cdot)$ corresponds to the synthetic network with the label-specific activation maps $\mathcal{C}$ as part of it. Next, we can train the multi-label predictor f_{θ_c} by minimizing the following loss:

$$\Omega_2(f_{\theta_s}, f_{\theta_c}) = \frac{1}{n_g} \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} ||f_{\theta_c}(\mathbf{z}_{ij}) - N(\mathbf{q}_i, \mathbf{q}_j)||_2^2 \tag{4}$$

where n_g is the number of generated samples, $f_{\theta_c}(\cdot)$ is the multi-label predictor network, $N(\mathbf{q}_i, \mathbf{q}_j)$ is the label confidence vector of $\mathbf{z}_{ij}$, which is the normalization of the sum of $\mathbf{q}_i$ and $\mathbf{q}_j$. We use the mean squared error as the classification loss. Unlike previous classification networks that take a batch of individual samples to optimize the network parameters, our FsPML-SF uses a batch of generated samples induced from pairwise samples and original samples to optimize the classifiers, due to the pairwise generation, the number of training samples can be largely enlarged, which greatly improve the performance, as our experiments will show.

3.2.3. Unified Framework and Optimization

We integrate the optimization of label confidence matrix $\mathbf{Q}$, synthetic network f_{θ_s} for generating new samples with associated label confidence vectors and multi-label classifier f_{θ_c} into a unified manner as follows:

$$min \quad f_{\theta_c}(\mathbf{X}, \mathbf{Q}) + \alpha\Omega_1(\mathbf{Q}) + \beta\Omega_2(f_{\theta_s}, f_{\theta_c}) \tag{5}$$

where the first term denotes the loss of the prediction on the original data, α and β are the trade-off parameters for the last two terms to keep the balance of the model.

The alternative optimization procedure is employed to jointly optimize $\mathbf{Q}$, f_{θ_s} and f_{θ_c}. For simplicity, we define the optimized label confidence matrix as $\mathbf{Q}^\tau$ in τ-th iteration and initialize the label confidence matrix $\mathbf{Q}^1$ as follows:

$$q_{ik}^1 = \begin{cases} \frac{1}{|\mathcal{Y}_i|}, & \text{if } k \in \mathcal{Y}_i \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

We firstly perform label discrimination by viewing f_{θ_s} and f_{θ_c} as fixed, then Eq. (5) with respect to $\mathbf{Q}^\tau$ is reduced to:

$$\begin{aligned} min \ & \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x}_i \in \mathcal{D}} ||f_{\theta_c}^{(\tau-1)}(\mathbf{x}_i) - \mathbf{q}_i^\tau||_2^2 \\ & + \alpha\Big(\sum_{\mathbf{x}_i \in \mathcal{D}} \sum_{k \in \mathcal{Y}_i} (q_{ik}^\tau - \sum_{\mathbf{x}_j \in \mathcal{D}, j \neq i} [\![q_{jk} \neq 0]\!] s_{ij}^\tau t_{ij}^\tau q_{jk}^\tau)^2 \\ & + \lambda \sum_{\mathbf{x}_i \in \mathcal{D}} \sum_{k \in \mathcal{Y}_i} \sum_{h \in \mathcal{Y}_i, h \neq k} [\![q_{ik} q_{ih} \neq 0]\!] (q_{ik}^\tau q_{ih}^\tau - p_{kh}^\tau)^2\Big) \end{aligned} \tag{7}$$

Algorithm 1 FsPML-SF: Few-shot Partial Multi-label Learning with Synthetic Features Network

Input: the N-way K-shot training dataset $\mathcal{D}$; the max iteration T_1; the trade-off parameters α, β and λ

Output: confidence matrix $\mathbf{Q}$; the synthetic features network f_{θ_s}; multi-label classifier f_{θ_c}

Process:

1: Initialize the label confidence matrix $\mathbf{Q}$ via Eq. (6)
2: Setup feature similarity matrix $\mathbf{S}$ via Eq. (8)
3: **for** $\tau = 1 \rightarrow T_1$ **do**
4: Update the semantic similarity $\mathbf{T}$ and the label co-occurrence matrix $\mathbf{P}$ via Eqs. (9 -10)
5: Update $\mathbf{Q}^\tau$ via Eq. (7)
6: Fix $\mathbf{Q}^\tau$, generate synthetic samples via f_{θ_s} and compute the loss of via Eq. (4)
7: Compute the loss via Eq. (5) and update the net parameters of $f_{\theta_s}^\tau$, $f_{\theta_c}^\tau$ via Adam.
8: **end for**

where $\mathcal{D}$ denotes the set of training samples. $f_{\theta_c}^{(\tau-1)}(\mathbf{x}_i)$ is the predicted label vector of $\mathbf{x}_i$ in the last iteration, which is a constant for $\mathbf{Q}^\tau$. To quantify the feature and semantic similarities between samples, we adopt the widely-used cosine similarity. The feature similarity is computed as follows:

$$s_{ij}^\tau = \frac{\mathbf{x}_i^T \mathbf{x}_j}{||\mathbf{x}_i||_2 \cdot ||\mathbf{x}_j||_2} \tag{8}$$

Since $\mathbf{Q}$ embodies more credible label information than the original candidate label space $\mathbf{Y}$, the semantic similarity is calculated based on $\mathbf{Q}^{(\tau-1)}$ as follows:

$$t_{ij}^\tau = \frac{\mathbf{q}_i^{(\tau-1)T} \mathbf{q}_j^{(\tau-1)}}{||\mathbf{q}_i^{(\tau-1)}||_2 \cdot \ ||\mathbf{q}_j^{(\tau-1)}||_2} \tag{9}$$

We want to remark that other similarity metrics can also be adopted here, and our choice of cosine similarity is for its simplicity and wide application.

To capture the label co-occurrence as well as consider their credibility, $\mathbf{P}$ is updated based on $\mathbf{Q}^{(\tau-1)}$ in last iteration as:

$$p_{kh}^\tau = \frac{1}{n_{kh}} \sum_{\mathbf{x}_i \in \mathcal{D}} q_{ik}^{(\tau-1)} q_{ih}^{(\tau-1)} \tag{10}$$

where n_{kh} is the number of samples whose candidate label set simultaneously contains both label k and h. Based on the above definitions, we apply Quasi-Newton method to update $\mathbf{Q}^\tau$.

With the fixed $\mathbf{Q}^\tau$, our synthetic network f_{θ_s} and multi-label classifier $f_{\theta_c}^\tau$ are updated together by the canonically-used Adam optimizer [51].

Algorithm 1 summarizes the overall procedure of FsPML-SF.

4. Experiments

4.1. Experimental Setup

4.1.1. Datasets

To date, there are no off-the-shelf FsPML datasets for experiments. Following [15, 16, 19], we conduct experiments on two MML dataset benchmarks (**MS-COCO** [52] and **NUS-WIDE** [53]) with the following controlling ways. Specifically, we divide the multi-label dataset into two subsets $\mathcal{D}_{base}$ and $\mathcal{D}_{novel}$, each with a large number of base labels and a small number (N) of novel labels, respectively. We randomly sampled a number of few-shot partial multi-label sets with N classes from $\mathcal{D}_{base}$ to train the synthetic features networks f_{θ_s} and multi-label classifier f_{θ_c}. While in the evaluation phase, the few-shot partial multi-label sets are selected at random from $\mathcal{D}_{novel}$.

MS-COCO is constructed for image recognition, segmentation, and caption and it has recently been employed to evaluate multi-label image classification. The dataset consists of 123000 images with 80 common labels. We adopt the COCO 2014 train and validation sets. Following the previous works [15, 19], the 80 labels are split into 64 base and 16 novel labels. Specifically, the novel labels are *bicycle, boat, stop sign, bird, backpack, frisbee, snowboard, surfboard, cup, fork, spoon, broccoli, chair, keyboard, microwave* and *vase*.

NUS-WIDE is a public multi-label image dataset which contains 269,648 images and these images are further manually annotated with 81 categories by human annotators. The 81 labels are split into 61 base ones and 20 novel ones following the previous work [19]. Specially, the novel labels are *airport, boats, bridge, cars, dog, garden, horses, house, lake, mountain, person, plane, plants, snow, street, train, tree, vehicle, wedding* and *window*.

Each few-shot PML dataset consists of samples containing only one or more of the N target labels. During the $\mathcal{D}_{base}$ training phase, we randomly chose N target labels from base labels with the guarantee of every label appearing K_1 times. When it comes to evaluation, the N target labels are selected from the novel ones and the dataset ensures K_2 examples per label. Due to the random selection when composing few-shot PML dataset, this balance is not always possible, and hence in some tasks, the number of samples per label could exceed by one at most. With the selected FsPML datasets, following the widely-used protocol for introducing irrelevant labels [1, 9, 19], we utilize use parameter p to control the proportion of samples with irrelevant labels, and parameter r to denote there are r noisy labels per PML sample, which are randomly selected from the label space of corresponding task.

4.1.2. Comparing Methods

To validate the effectiveness of FsPML-SF, we compare it against with FsPML [19], seven representative PML algorithms (PML-fp [1], fPML [2], PML-MAP [8], PML-NI [23], HALE [30], PML-LCom [11] and NATAL [13]) and two representative FsMLL solutions (LaSO[15] and KGGR [16]). Each comparison method is configured with the suggested parameters in the corresponding papers or codes.

- **FsPML** [19] first performs adaptive distance metric learning via an embedding network using both sample features and label semantics in the embedding space. Next it

rectifies the positive and negative prototypes of each new label of the target task in the embedding space. Suggested configuration: learning rate $lr = 0.000001$, trade-off parameters $\lambda_1 = 0.6$, $\lambda_2 = 0.1$, the number of nearest neighbors in label semantics $k_2 = 1$, the number of iterations for rectifying prototype $iter = 5$.

- **PML-fp** [1] mainly minimizes a rank loss weighted by the confidences and exploits structure information in feature space to optimize the ground-truth condidence of candidate labels. Suggested configuration: trade-off parameters $C_1 = 1$ and $C_3 = 10$.

- **fPML** [2] builds on low-rank assumption of the label matrix and utilizes the coherence between the label and feature data matrix to estimate the label confidence. Suggested configuration: trade-off parameters $\lambda_1 = 0.1$, $\lambda_2 = 1$ and $\lambda_3 = 0.1$.

- **PML-MAP** [8] is a two-stage method that estimates credible labels via label propagation first and then induces multi-label classifier. Suggested configuration: trade-off parameters $\alpha = 0.95$ and credible label elicitation threshold $thr = 0.9$.

- **PML-NI** [23] is based on the relationships between noisy labels and feature contents. It simultaneously recover the ground-truth information and identify the noisy labels. Suggested configuration: trade-off parameters $\lambda = 1$, $\beta = 1$ and $\delta = 0.5$.

- **HALE** [30] interprets label disambiguation as instance-to-label matchings and formulates the task of PML as a matching selection problem. The problem is solved by utilizing Graph Matching scheme with many-to-many constraint. Suggested configuration: the number of nearest neighbors in prediction $k = 10$.

- **PML-LCom** [11] factorizes the relevant label matrix into two low-rank matrices. Then it optimizes the coefficient matrix of the multi-label predictor with respect to the compressed label matrix. Suggested configuration: trade-off parameters $\lambda_1 = 5$, $\lambda_2 = 10$ and λ_3=0.1.

- **NATAL** [13] introduces a "missing" feature matrix and constrains both the "missing" feature matrix and the prediction parameter matrix to be low-rank. Then it optimizes them in an alternative manner. Suggested configuration: trade-off parameters $\alpha = 1$, $\beta = 10^{-6}$ and $\lambda = 10^{-2}$.

- **LaSO**[15] mainly tackles the few-shot problem via data augmentation in feature space and generates new multi-label samples by combining other samples. Suggested configuration: learning rate $lr = 0.001$, training epochs $epoch = 40$, and union-based augmentation model as the best performance model among union, subtraction and intersection, is used to perform data augmentation.

- **KGGR**[16] mainly exploits prior knowledge to guide adaptive information propagation among different categories to facilitate multi-label analysis and reduce the dependency of training samples. Suggested configuration: in stage 1, learning rate $lr = 0.00001$ and training epochs $epoch = 20$, while in stage 2, learning rate $lr = 0.0001$, training iterations $iteration = 500$ and trade-off parameters $\delta = 0.001$.

As to our **FsPML-SF**, the trade-off parameters $\alpha = 10$, $\beta = 1$, $\lambda = 0.1$, the number of iterations T_1 during the $\mathcal{D}_{base}$ training phase is fixed to 50, and the number of iterations T_2 during the $\mathcal{D}_{novel}$ training phase is fixed to 5. To better train the networks, for the few-shot PML dataset from $\mathcal{D}_{base}$, we generate $\binom{n}{2}$ synthetic samples. While for the few-shot PML dataset from $\mathcal{D}_{novel}$, to maintain efficacy, we generate $\lfloor \binom{n_2}{2}/2\dot{K}_2 \rfloor$ synthetic samples, where n_2 is the number of original few-shot samples within a task of $\mathcal{D}_{novel}$. In addition, the InceptionV3 [50] pretrained on the ImageNet [54] is utilized as feature extractor backbone and Adam [51] optimizer is applied to optimize network parameters. For fair comparison, FsPML, LaSO and KGGR also use the InceptionV3 pretrained on ImageNet as the initial backbone network. For many-shot PML methods, they use the image features extracted by InceptionV3 pretrained on ImageNet and $\mathcal{D}_{novel}$ for training.

4.1.3. Evaluation Metrics

To make a comprehensive performance evaluation and comparison, we employ six canonically used multi label evaluation metrics [3] to estimate each compared method, including *Ranking Loss*, *One-Error*, *Coverage*, Mean Average Precision (*MAP*), *Macro-F1*, and *Micro-F1*. For *Ranking Loss*, *One-Error* and *Coverage*, the smaller the value, the better the performance is; while for *MAP Macro-F1* and *Micro-F1*, the larger the value, the better the performance is. These metrics quantify the performance from different perspectives, and it is difficult for an approach consistently outperforming the others across all the metrics.

4.2. Results Analysis

4.2.1. Results on MS COCO

The detailed experiments results (10 independent runs) of each compared method on MS COCO with $p = 0.8$, $r \in \{1, 2, 3\}$, $K_2 \in \{5, 10\}$ are reported in Table 2 and Tabel 3. From this Table, we have the following observations:

(i) For each experiment setting and evaluation metric, FsPML-SF achieves a better performance than other compared methods in almost all cases. These superior results prove the effectiveness of FsPML-SF on few-shot PML samples.

(ii) **Compared methods** vs. {K_2,r}: We observe that under a fixed r, nearly all the methods' performance improves when K_2 increases from 5 to 10. This is because more training samples are available to induce the classifier and this phenomena justifies that few-shot training samples indeed degenerate the performance. On the other hand, with the increase of r under a fixed K_2, it's inevitable that all the methods have a reduced performance due to more irrelevant labels of PML samples. This observation indicates the importance of handling few-shot and PML samples and again demonstrates the effectiveness of FsPML-SF.

(iii)**FsPML-SF** vs. **FsPML**: These two methods both target to tackle the few-shot PML problem, FsPML mainly borrows the idea of prototype network in a meta learning manner to combat with scarce few-shot data, and FsPML-SF resorts to data augmentation by synthetic network with label confidence induced features. On the whole, our FsPML-SF achieves a better performance than FsPML, which is potentially limited by the projected prototypes that fails to mine the feature and label information in a sensible way. While FsPML-SF

Table 2: Results (mean±std) of each method in terms of *Ranking Loss*, *One Error* and *Coverage* on **MS COCO**. K_2: The number of training samples per class in the FsPML dataset; r: the number of irrelevant labels per PML sample; The proportion of noisy samples p in support set is set to be 80%. $\circ/\bullet$ indicates that FsPML-SF is statistically worse/better than the compared method, and the statistical significance is assessed by student pairwise t-test at 95% confident level.

	Ranking Loss↓		One Error↓		Coverage↓	
	$K_2 = 5$	$K_2 = 10$	$K_2 = 5$	$K_2 = 10$	$K_2 = 5$	$K_2 = 10$
	$r = 1$					
FsPML-SF	.147±.003	.114±.007	.314±.033	.282±.046	.251±.014	.211±.011
FsPML	.220±.019•	.179±.020•	.462±.052•	.433±.051•	.326±.027•	.273±.027•
PML-fp	.283±.062•	.252±.033•	.562±.089•	.495±.038•	.411±.057•	.370±.037•
fPML	.161±.003•	.128±.029	.388±.050•	.333±.045•	.268±.038	.226±.041
PML-MAP	.316±.059•	.188±.049•	.528±.062•	.436±.052•	.459±.038•	.316±.057•
PML-NI	.167±.018•	.159±.044•	.369±.065•	.372±.076•	.283±.022•	.267±.051•
HALE	.312±.051•	.311±.034•	.436±.057•	.367±.051•	.432±.037•	.376±.024•
PML-LCom	.262±.025•	.251±.026•	.601±.071•	.576±.029•	.378±.028•	.371±.034•
NATAL	.155±.002•	.119±.023	.353±.036•	.327±.067	.271±.023•	.217±.028
LaSO	.217±.018•	.183±.017•	.533±.046•	.500±.031•	.322±.020•	.276±.018•
KGGR	.212±.016•	.174±.019•	.517±.064•	.495±.038•	.353±.023•	.304±.052•
	$r = 2$					
FsPML-SF	.180±.017	.171±.019	.405±.042	.392±.068	.294±.030	.275±.021
FsPML	.237±.037•	.191±.024•	.501±.076•	.463±.048•	.342±.043•	.285±.029
PML-fp	.322±.058•	.309±.040•	.589±.112•	.544±.078•	.442±.043•	.419±.031•
fPML	.185±.026	.175±.018	.459±.061•	.401±.046	.299±.029	.284±.025
PML-MAP	.348±.036•	.286±.023•	.591±.084•	.408±.063	.463±.049•	.326±.038•
PML-NI	.194±.020	.172±.013	.483±.081•	.398±.043	.307±.023	.291±.019•
HALE	.358±.049•	.351±.052•	.507±.059•	.454±.055•	.467±.030•	.395±.028•
PML-LCom	.316±.032•	.270±.025•	.627±.058•	.619±.047•	.437±.032•	.386±.027•
NATAL	.190±.019	.182±.018•	.432±.061	.393±.066	.306±.022	.275±.023
LaSO	.238±.017•	.224±.014•	.573±.032•	.527±.052•	.346±.016•	.327±.016•
KGGR	.227±.046•	.195±.017•	.542±.060•	.513±.031•	.365±.019•	.304±.015•
	$r = 3$					
FsPML-SF	.210±.035	.181±.019	.452±.098	.439±.066	.332±.045	.294±.029
FsPML	.263±.040•	.186±.021	.536±.058•	.492±.041•	.375±.042•	.287±.022
PML-fp	.369±.042•	.315±.015•	.665±.091•	.640±.057•	.479±.041•	.424±.019•
fPML	.214±.029	.194±.018	.549±.084•	.462±.046	.320±.034	.298±.028
PML-MAP	.341±.042•	.274±.023•	.616±.077•	.514±.065•	.492±.048•	.332±.039•
PML-NI	.220±.038	.224±.023•	.530±.086•	.489±.055•	.336±.039	.346±.019•
HALE	.443±.021•	.429±.033•	.532±.082•	.493±.047•	.486±.041•	.415±.022•
PML-LCom	.331±.019•	.323±.041•	.674±.045•	.653±.044•	.428±.023•	.416±.026•
NATAL	.216±.031	.203±.011•	.501±.087	.465±.046	.328±.035	.288±.019
LaSO	.274±.021•	.229±.017•	.638±.017•	.545±.027•	.381±.022•	.336±.020•
KGGR	.265±.042•	.211±.031•	.626±.007•	.523±.045•	.402±.033•	.334±.014•

Table 3: Performance (mean±std) of comparing methods in terms of *MAP*, *Macro F1* and *Micro F1* on **MS COCO**. K_2: The number of training samples per class in the FsPML dataset; r: the number of irrelevant labels per PML sample; The proportion of noisy samples p in support set is set to be 80%. ∘/• indicates that FsPML-SF is statistically worse/better than the compared method, and the statistical significance is assessed by student pairwise t-test at 95% confident level.

	MAP↑		Macro F1↑		Micro F1↑	
	$K_2 = 5$	$K_2 = 10$	$K_2 = 5$	$K_2 = 10$	$K_2 = 5$	$K_2 = 10$
	$r = 1$					
FsPML-SF	.601±.026	.629±.031	.458±.019	.453±.057	.451±.023	.447±.051
FsPML	.557±.032•	.588±.037•	.384±.025•	.428±.021	.387±.025•	.425±.018•
PML-fp	.382±.049•	.422±.040•	.322±.067•	.352±.045•	.326±.068•	.354±.034•
fPML	.530±.033•	.569±.039•	.439±.027•	.450±.045	.425±.021•	.442±.029
PML-MAP	.461±.035•	.548±.047•	.202±.034•	.234±.007•	.189±.016•	.199±.007•
PML-NI	.501±.036•	.552±.058•	.352±.037•	.388±.059•	.369±.034•	.401±.027•
HALE	.449±.046•	.508±.038•	.328±.035•	.408±.029•	.375±.027•	.409±.043•
PML-LCom	.482±.034•	.510±.049•	.341±.013•	.360±.031•	.301±.031•	.326±.033•
NATAL	.466±.036•	.511±.075•	.437±.021•	.446±.037	.421±.029•	.441±.016
LaSO	.414±.027•	.460±.023•	.209±.037•	.242±.016•	.215±.031•	.237±.027•
KGGR	.418±.008•	.441±.020•	.207±.026•	.254±.031•	.201±.019•	.241±.032•
	$r = 2$					
FsPML-SF	.557±.028	.589±.017	.379±.039	.440±.057	.381±.032	.421±.015
FsPML	.519±.053•	.571±.031	.376±.039	.412±.013	.364±.019	.402±.028
PML-fp	.378±.057•	.392±.053•	.276±.083•	.299±.056•	.283±.040•	.307±.035•
fPML	.488±.025•	.557±.032•	.375±.028•	.407±.039•	.370±.028	.397±.021•
PML-MAP	.426±.068•	.535±.045•	.198±.045•	.203±.007•	.198±.033•	.206±.011•
PML-NI	.449±.024•	.499±.023•	.297±.049•	.375±.042•	.313±.035•	.384±.024•
HALE	.425±.035•	.472±.033•	.252±.043•	.382±.031•	.354±.012•	.377±.027•
PML-LCom	.419±.045•	.474±.031•	.276±.038•	.291±.041•	.289±.011•	.295±.025•
NATAL	.461±.041•	.477±.092•	.377±.022	.414±.027•	.380±.011	.409±.012
LaSO	.368±.021•	.408±.020•	.193±.024•	.224±.033•	.211±.013•	.227±.023•
KGGR	.398±.027•	.409±.020•	.203±.016•	.236±.052•	.193±.021•	.228±.019•
	$r = 3$					
FsPML-SF	.534±.043	.572±.027	.377±.069	.417±.037	.373±.016	.405±.023
FsPML	.481±.044•	.565±.024	.338±.029•	.409±.021	.351±.020•	.397±.029
PML-fp	.374±.038•	.411±.038•	.237±.048•	.302±.016•	.199±.045•	.246±.030•
fPML	.438±.043•	.508±.023•	.368±.030	.405±.029	.367±.019•	.402±.027
PML-MAP	.418±.052•	.548±.023•	.231±.067•	.200±.006•	.236±.017•	.203±.010•
PML-NI	.407±.056•	.425±.024•	.265±.054•	.311±.021•	.263±.027•	.324±.025•
HALE	.368±.043•	.443±.041•	.227±.044•	.368±.028•	.334±.042•	.364±.034•
PML-LCom	.399±.031•	.418±.058•	.254±.029•	.271±.039•	.249±.026•	.265±.021•
NATAL	.446±.053•	.470±.046•	.371±.019	.383±.025•	.372±.014	.379±.016•
LaSO	.324±.027•	.373±.022•	.183±.020•	.180±.037•	.191±.034•	.201±.015•
KGGR	.279±.029•	.365±.040•	.179±.065•	.201±.017•	.175±.033•	.206±.023•

Table 4: Results (mean±std) of each method in terms of *Ranking Loss*, *One Error* and *Coverage* on **NUS-WIDE**. K_2: The number of training samples per class in the FsPML dataset; r: the number of irrelevant labels per PML sample; The proportion of noisy samples p in support set is set to be 80%. $\circ/\bullet$ indicates that FsPML-SF is statistically worse/better than the compared method, and the statistical significance is assessed by student pairwise t-test at 95% confident level.

	Ranking Loss↓		One Error↓		Coverage↓	
	$K_2 = 5$	$K_2 = 10$	$K_2 = 5$	$K_2 = 10$	$K_2 = 5$	$K_2 = 10$
	$r = 1$					
FsPML-SF	.170±.041	.152±.017	.465±.094	.448±.048	.252±.045	.222±.019
FsPML	.291±.011•	.203±.018•	.602±.037•	.570±.013•	.362±.015•	.275±.021•
PML-fp	.341±.037•	.296±.003•	.609±.076•	.576±.067•	.470±.029•	.422±.026•
fPML	.212±.033•	.158±.018	.545±.048•	.457±.044	.310±.033•	.229±.012
PML-MAP	.354±.051•	.248±.024•	.713±.069•	.640±.051•	.437±.034•	.362±.018•
PML-NI	.208±.047•	.171±.018•	.549±.075•	.489±.071•	.302±.045•	.251±.016•
HALE	.392±.049•	.322±.047•	.612±.020•	.511±.025•	.439±.033•	.376±.028•
PML-LCom	.316±.039•	.269±.037•	.741±.074•	.694±.040•	.406±.045•	.346±.048•
NATAL	.226±.038•	.199±.028•	.563±.068•	.499±.047•	.313±.034•	.264±.033•
LaSO	.304±.028•	.250±.031•	.745±.060•	.670±.033•	.392±.026•	.320±.038•
KGGR	.284±.012•	.236±.016•	.732±.027•	.669±.034•	.359±.023•	.279±.021•
	$r = 2$					
FsPML-SF	.189±.028	.176±.033	.505±.056	.503±.042	.260±.035	.254±.035
FsPML	.282±.021•	.225±.013•	.613±.129•	.601±.102•	.371±.029•	.301±.017•
PML-fp	.375±.022•	.342±.017•	.682±.066•	.658±.007•	.502±.015•	.447±.022•
fPML	.238±.024•	.187±.018	.613±.034•	.538±.027•	.331±.032•	.262±.026
PML-MAP	.367±.037•	.343±.038•	.714±.055•	.646±.029•	.461±.032•	.418±.037•
PML-NI	.220±.019•	.188±.025	.570±.012•	.544±.047•	.314±.025•	.273±.029
HALE	.408±.059•	.345±.046•	.647±.010•	.556±.057•	.480±.042•	.382±.032•
PML-LCom	.337±.029•	.333±.045•	.798±.052•	.777±.054•	.435±.030•	.429±.047•
NATAL	.221±.017•	.223±.028•	.556±.031•	.520±.014	.318±.019•	.288±.028•
LaSO	.316±.034•	.284±.035•	.753±.040•	.732±.067•	.409±.046•	.370±.034•
KGGR	.289±.050•	.254±.034•	.747±.016•	.701±.046•	.373±.018•	.297±.035•
	$r = 3$					
FsPML-SF	.223±.027	.215±.051	.619±.038	.606±.057	.307±.036	.271±.044
FsPML	.279±.024•	.253±.012•	.670±.093•	.630±.081	.377±.029•	.324±.018•
PML-fp	.404±.027•	.303±.021•	.713±.094•	.662±.012•	.525±.029•	.481±.026•
fPML	.249±.029•	.219±.015	.687±.041•	.612±.066	.338±.032•	.277±.019
PML-MAP	.405±.012•	.371±.011•	.704±.038•	.639±.039	.499±.074•	.429±.054•
PML-NI	.239±.018•	.216±.017	.660±.033•	.605±.056	.327±.032•	.295±.022
HALE	.417±.038•	.388±.021•	.683±.023•	.627±.030	.491±.031•	.422±.009•
PML-LCom	.356±.035•	.339±.036•	.804±.064•	.790±.032•	.449±.041•	.419±.031•
NATAL	.231±.009	.218±.025	.629±.033	.545±.062	.314±.015	.272±.028
LaSO	.323±.027•	.315±.032•	.766±.024•	.761±.033•	.414±.033•	.397±.034•
KGGR	.315±.011•	.298±.041•	.753±.046•	.724±.040•	.409±.032•	.362±.017•

Table 5: Performance (mean±std) of comparing methods in terms of *MAP*, *Macro F1* and *Micro F1* on **NUS-WIDE**. K_2: The number of training samples per class in the FsPML dataset; r: the number of irrelevant labels per PML sample; The proportion of noisy samples p in support set is set to be 80%. ∘/• indicates that FsPML-SF is statistically worse/better than the compared method, and the statistical significance is assessed by student pairwise t-test at 95% confident level.

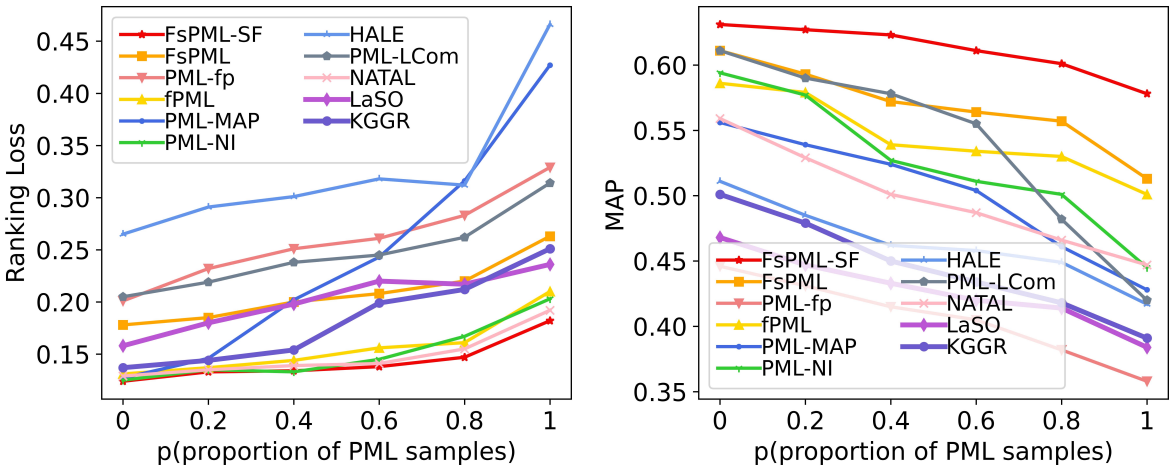
	MAP↑		Macro F1↑		Micro F1↑	
	$K_2 = 5$	$K_2 = 10$	$K_2 = 5$	$K_2 = 10$	$K_2 = 5$	$K_2 = 10$
	$r = 1$					
FsPML-SF	.557±.081	.606±.031	.327±.075	.371±.020	.325±.043	.366±.017
FsPML	.433±.007•	.508±.011•	.217±.017•	.359±.019	.225±.023•	.363±.019
PML-fp	.275±.047•	.316±.019•	.255±.028•	.276±.018•	.261±.029•	.277±.026•
fPML	.423±.038•	.475±.026•	.326±.025	.348±.041•	.323±.027	.346±.040
PML-MAP	.287±.023•	.371±.019•	.137±.007•	.144±.006•	.136±.011•	.144±.005•
PML-NI	.413±.068•	.484±.037•	.199±.041•	.304±.037•	.206±.027•	.335±.013•
HALE	.337±.042•	.386±.036•	.231±.034•	.290±.041•	.299±.021	.337±.034•
PML-LCom	.348±.046•	.430±.035•	.212±.039•	.254±.043•	.242±.037•	.301±.024•
NATAL	.246±.059•	.274±.067•	.271±.033•	.271±.023•	.266±.030•	.274±.023•
LaSO	.351±.055•	.407±.049•	.163±.027•	.209±.026•	.205±.034•	.228±.029•
KGGR	.365±.031•	.412±.038•	.162±.044•	.220±.028•	.246±.011•	.303±.019•
	$r = 2$					
FsPML-SF	.549±.061	.579±.038	.313±.044	.351±.032	.317±.035	.363±.027
FsPML	.413±.017•	.479±.031•	.216±.025•	.350±.023	.303±.043	.336±.025•
PML-fp	.253±.035•	.267±.003•	.184±.044•	.178±.011•	.193±.043•	.207±.009•
fPML	.375±.054•	.465±.031•	.311±.029	.388±.023∘	.302±.027	.371±.018∘
PML-MAP	.321±.013•	.371±.037•	.147±.021•	.142±.005•	.148±.021•	.143±.005•
PML-NI	.382±.043•	.423±.041•	.195±.029•	.262±.036•	.194±.013•	.291±.043•
HALE	.306±.032•	.371±.041•	.180±.033•	.258±.037•	.225±.039•	.327±.035•
PML-LCom	.337±.046•	.356±.022•	.203±.043•	.243±.043•	.234±.032•	.257±.037•
NATAL	.230±.072•	.271±.049•	.268±.027•	.274±.013•	.264±.034•	.276±.009•
LaSO	.325±.051•	.352±.045•	.147±.030•	.197±.030•	.201±.018•	.223±.023•
KGGR	.335±.031•	.399±.020•	.162±.007•	.206±.019•	.243±.021•	.279±.017•
	$r = 3$					
FsPML-SF	.494±.053	.512±.035	.257±.040	.341±.031	.271±.034	.335±.028
FsPML	.387±.022•	.432±.015•	.239±.017	.308±.018•	.255±.021	.313±.017
PML-fp	.246±.038•	.252±.028•	.154±.014•	.241±.017•	.179±.026•	.257±.032•
fPML	.328±.031•	.423±.015•	.301±.038∘	.331±.012	.285±.023∘	.326±.017
PML-MAP	.303±.031•	.346±.038•	.136±.023•	.213±.004•	.137±.013•	.216±.021•
PML-NI	.331±.022•	.367±.029•	.149±.022•	.253±.038•	.163±.021•	.257±.026•
HALE	.282±.046•	.333±.036•	.172±.034•	.227±.019•	.250±.024	.309±.021•
PML-LCom	.310±.025•	.337±.048•	.201±.025•	.229±.033•	.223±.035•	.231±.024•
NATAL	.223±.060•	.256±.059•	.228±.012•	.249±.013•	.245±.014•	.231±.009•
LaSO	.296±.033•	.312±.025•	.142±.028•	.153±.013•	.157±.031•	.193±.023•
KGGR	.328±.023•	.356±.024•	.143±.065•	.194±.009•	.231±.018•	.264±.032•

utilizes PML samples from diverse aspects for disambiguating labels and to generate high-quality samples to remedy the lack of PML samples in a sensible way, thus it gives a better performance.
(iv) **Few-shot** vs. **many-shot PML**: FsPML-SF and FsPML are superior to the other PML methods in most cases and the performance gap is more prominent when $K_2 = 5$. Despite no explicit requirement for the number of training samples is claimed in these compared PML methods, they still fail to generalize well with few-shot data. This commendably proves that existing many-shot PML methods are heavily limited by the number of training sample and signifies the importance to tackle PML in few-shot scenarios. We note NATAL often have the best performance among the many-shot solutions. This pattern suggests the noisy labels of images may be caused by corrupted features. Even though, FsPML-SF still improves NATAL by 30% in MAP and 33.3% in MacroF1.
(v) **With** vs. **Without modelling noisy labels**: The FsMLL methods (LaSO and KGGR) target for few-shot multi-label data, but they do not explicitly take the irrelevant labels of training data into consideration. Thus, they are clearly outperformed by FsPML-SF and FsPML, which concretely model the incorrect labels of multi-label data. Alike LaSO, FsPML-SF also augments the multi-label samples to enlarge the training data, but it considers the irrelevant labels of training samples and augment training samples in a more credible way by extracting label informative features using updated label confidences. As a result, FsPML-SF on average improves LaSO by 57.2% in terms of these six evaluation metrics. These results prove the necessity to account for irrelevant labels and show the vulnerability of FsMLL methods when dealing with noisy few-shot PML samples.

4.2.2. Results on NUS-WIDE

To better demonstrate the experimental phenomenon, we also conduct experiments on another dataset NUS-WIDE with teh same control setting. Table 4 and Table 5 report the results of each compared method on NUS-WIDE with $p = 0.8$, $r \in \{1, 2, 3\}$, $K_2 = \{5, 10\}$. FsPML-SF again clearly outperforms other compared methods across all evaluation metrics, and the conclusions are similar as those on MS COCO. Each method has a lower performance on NUS-WIDE than on MS COCO, that is because each image of NUS-WIDE on average is annotated with more labels than that of MS COCO and there are more novel labels in the target tasks, which increases the task difficulty. However, our FsPML-SF is much less impacted than other compared methods, for its leverage of feature similarity, label similarity, updated label confidence vectors and label co-occurrence in a principle way. FsPML-SF gives significantly better results than the most related LaSO and FsPML, which again confirm the advantage of our label disambiguation and data augmentation strategy.

In addition, we used the signed-rank test to check the statistical difference between FsPML-SF and other compared methods across these metrics and datasets, all the p-value are smaller than 0.001.



(a) Ranking Loss↓ on MS COCO (b) MAP↑ on MS COCO

Figure 4: Performance of comparison methods in terms of *Ranking Loss* and *MAP* on **MS COCO** with different proportion PML samples ($K_2 = 5$ and $r = 1$).

4.3. Further Analysis

4.3.1. Negative Impacts of PML Samples

We conduct experiments on each compared methods with different proportions of PML samples to demonstrate the negative impact of PML samples. Specifically, we change the proportion of PML samples p from 0 to 1 with an interval of 0.2 on MS COCO with $r = 1$ and $K_2 = 5$. The results of each compared approach in terms of *Ranking Loss* and *MAP* are revealed in Figure 4.

Form the figure, we have the following observations:

(i) **Under** $p = 0$: When $p = 0$, each sample is precisely annotated and thus turn the problem into a few-shot multi-label classification task. Compared with other methods, our FsPML-SF still indicates a better or equal performance, which demonstrates that our proposed FsPML-SF effectively performs data augmentation and is capable of tackling the standard few-shot multi-label classification problem.

(ii) **Varying** p: With the increase proportion of PML samples, all the compared methods achieve a lower performance for the increased difficulty of label de-noising. Moreover, the degenerated performance of compared method is more notable as p gets higher. However, our FsPML-SF is much less impacted than other compared methods, for its proper leverage of feature similarity, label similarity and label co-occurrence in label disambiguation. These better results justify that our FsPML-SF is more noise-robust than those methods.

(iii) **Under** $p = 1$: When $p = 1$, which implies that all the samples are assigned with noisy annotations. Thus, it's more difficult to handle classification task in this scenario. Even though, our FsPML-SF still achieve a better performance than other compared methods, which again proves that the effectiveness and noise-robustness of our method..

4.3.2. Ablation Study

To further analyze the contribution factors of FsPML-SF, we conduct ablation experiments. For this purpose, we introduce four variants of FsPML-SF: **FsPML-SF(nS)**, **FsPML-SF(nT)**, **FsPML-SF(nP)**, **FsPML-SF(nF)**. The first three variants separately exclude the feature similarity, semantic similarity, label co-occurrences in Eq. (1) when updating the label confidence matrix $\mathbf{Q}$; while FsPML-SF(nF) does not utilize the label-confidence aware vectors in the synthetic network but the original feature vectors. Fig. 5 reveals the results of four variants and the full model on MS COCO and NUS-WIDE with p=0.8, K_2=5 and r=1.

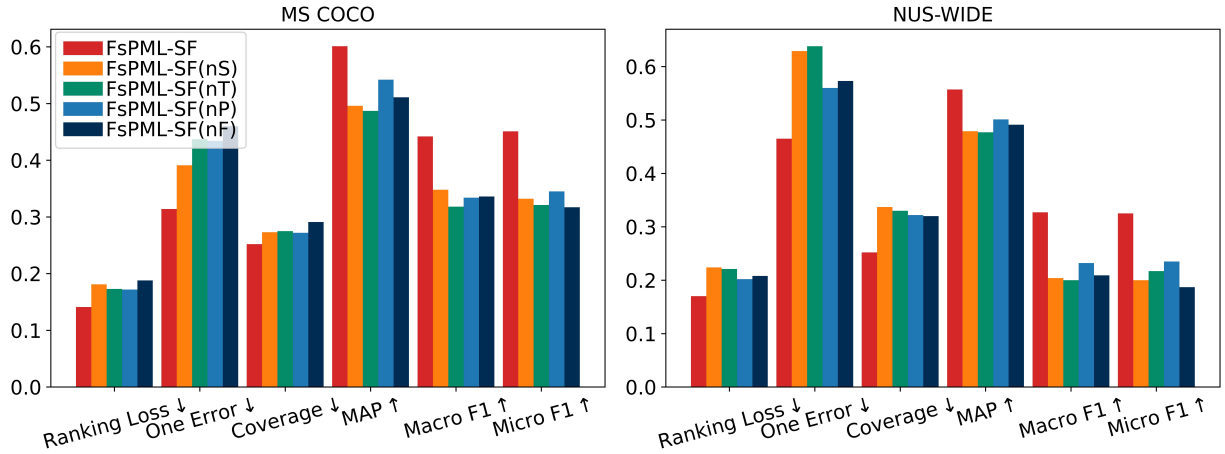


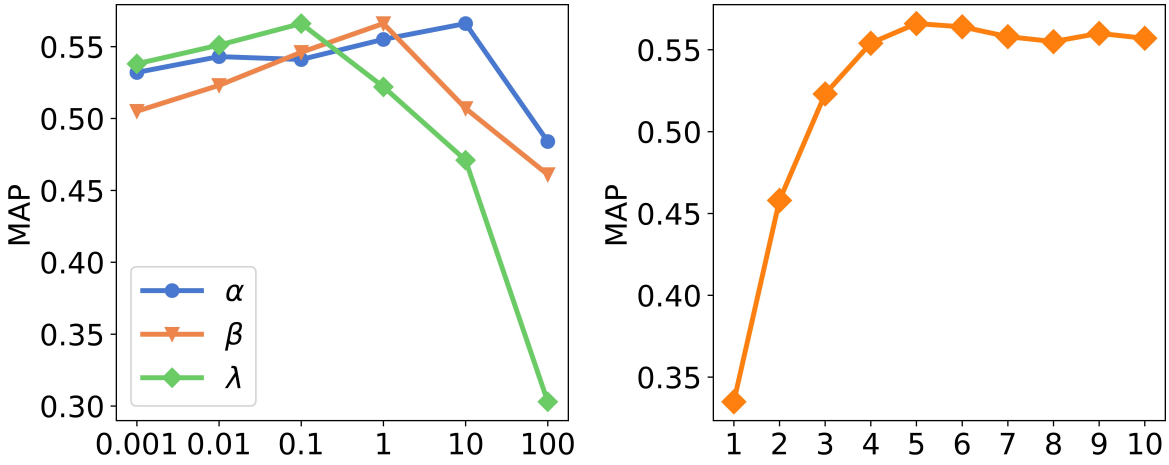
Figure 5: FsPML-SF vs. its degenerated variants on **MS COCO** and **NUS-WIDE** ($p = 0.8$, $K_2 = 5$ and $r = 1$). The smaller the value of the first three metrics is, the better the performance is, while the opposite holds for the last three (*MAP*, *MacroF1* and *MicroF1*)

(i) The full model FsPML-SF achieves a superior performance than its degenerated variants, which proves the rationality of FsPML-SF on disambiguating labels and generating samples with credible labels.
(ii) FsPML-SF(nS), FsPML-SF(nT) and FsPML-SF(nP) mainly quantify the contribution of feature similarity, semantic similarity and label co-occurrence for label discrimination, respectively. Following Fig. 5, we observe that any two of the three components can not lead to a comparable performance as the full model FsPML-SF. These results demonstrate the significance to proper utilize both the feature similarity, semantic similarity and label co-occurrence in label disambiguation.
(iii) FsPML-SF(nS) and FsPML-SF(nT) across the similar and lowest performance across the evaluation metrics, which corroborates that both the feature and label information more contribute to label disambiguation.
(iv) FsPML-SF(nP) outperforms other variants across most metrics, this indicates that label co-occurrence statistics have the least contribution in label disambiguation than other factors. The possible reason is that the overcomplete noisy annotations containing in the the candidate label set make it hard to effectively explore the label correlation. However, the performance gap between the full model FsPML-SF and FsPML-SF(nP) proves that the contribution of

label co-occurrence statistics is also non-trivial.
(v) There is a notable performance disparity between FsPML-SF and FsPML-SF(nF), which indicates the synthetic vectors of FsPML-SF can highlight the label-informative features with high confidence values and the generated samples are beneficial to induce the multi-label predictor.

4.3.3. Parameter Sensitivity Analysis

We study the parameter sensitivity of FsPML-SF *w.r.t.* the trade-off parameters α, β and λ, T_2 (the max number of iterations for training on $\mathcal{D}_{novel}$). We vary one trade-off parameter in the range of $\{0.001,\ 0.01, \cdots,\ 100\}$ while fixing the other two. We change T_2 from 1 to 10 with an interval of 1 and conduct experiments on MS COCO dataset with $p = 0.8$, $K_2 = 5$ and $r = 2$. The results in terms of *MAP* are shown in Fig. 6(a) and Fig. 6(b), respectively.



(a) MAP vs. α β, λ (b) MAP vs. T_2

Figure 6: Results of FsPML-SF under different input values of parameters vs. α, β and λ, T_2 in terms of *MAP* on **MS COCO** ($p = 0.8$, $K_2 = 5$ and $r = 2$)

(i) Sensitivity to α: From Fig. 6(a), when $\alpha \approx 10$, FsPML-SF achieves the best performance. This fact proves the necessity to use the refined labels of training samples, instead of the original ones, to generate new samples and to train the multi-label predictor. When α is too large or too small, the label discrimination is under-weighted or over-weighted, thus FsPML-SF has a reduced performance.
(ii) Sensitivity to β: As Fig. 6(a) shows, FsPML-SF achieves the best performance when $\beta \approx 1$. When β is too small, the loss of generated samples is not well accounted, which leads to a poor performance. When β is too large (i.e., ≥ 100), FsPML-SF also achieves a poor performance, as it excessively over-weights the synthetic features networks, but under-weights the prediction model and label disambiguation model.
(iii) Sensitivity to λ: λ balances the usage of label co-occurrence in label discrimination stage. As shown in Fig. 6(a), FsPML-SF manifests a degraded performance when λ is too small/large. This is because a too small λ nearly disregard the label co-occurrence in

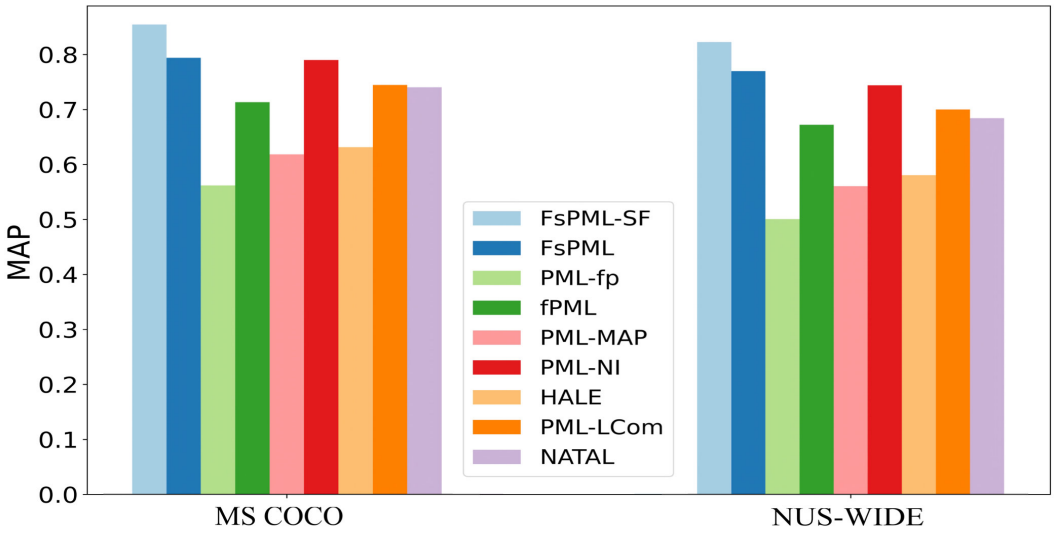


Figure 7: Performance *MAP* between disambiguated labels elicited by each PML methods and corresponding ground-truth labels on **MS COCO** and **NUS-WIDE** ($p = 0.8$ and $r = 2$)

denoising while a too large λ over-weights the approximated label correlation, both extreme cases give a negative impact on the optimization of $\mathbf{Q}$.
(iv)Sensitivity to T_2: In Fig. 6(b), the MAP value of FsPML-SF gradually increases and reaches to the maximum when $T_2 \approx 5$, and it maintains relatively stable when $T_2 \geq 5$. This pattern proves the efficacy of FsPML-SF for fast adapting to novel labels with few-shot samples.

4.3.4. Label Disambiguation Ability Analysis

To further analyze the label disambiguation ability of our FsPML-SF. We conduct experiments to verify the correlation between the disambiguated labels with their corresponding ground-truth ones. To provide a comparison, we compare FsPML-SF against eight representative PML methods. Specifically, we calculate the evaluation metric *MAP* between the disambiguated labels obtained by each PML methods with their valid annotations on MS COCO and NUS-WIDE dataset with $p = 0.8$, $K_2 = 5$ and $r = 2$. Fig.7 reveal the results.

We can obeserve that our FsPML-SF achieves a most successful performance compared with other PML methods. These superior results manifest our disambiguation ability to the candidate label set in few-shot setting. With only a handful of PML samples, our FsPML-SF makes a collaborative use of feature similarity, semantic similarity and label correlations and thus obtains label confidence values with high quality. FsPML-SF can effectively perform label disambiguation with limited samples and dislodge the irrelevant labels in a rational way.

4.3.5. Impact of Generated Samples

FsPML-SF introduces the synthetic feature network f_{θ_s} to generate new samples with label credibility scores and extract the label-informative features. To validate the effectiveness of f_{θ_s}, we conduct experiments on $\mathcal{D}_{novel}$ of NUS-WIDE by varying the number of generated

samples in the range of $\{ 0, \lfloor\binom{n_2}{2}/50\rfloor, \lfloor\binom{n_2}{2}/40\rfloor, \lfloor\binom{n_2}{2}/30\rfloor, \lfloor\binom{n_2}{2}/20\rfloor, \lfloor\binom{n_2}{2}/10\rfloor, \lfloor\binom{n_2}{2}/5\rfloor, \binom{n_2}{2}\}$ (n_2 is the number of original samples in each task). The results on NUS-WIDE with $p = 0.8$ and $r = 1$ are shown in Fig. 8.

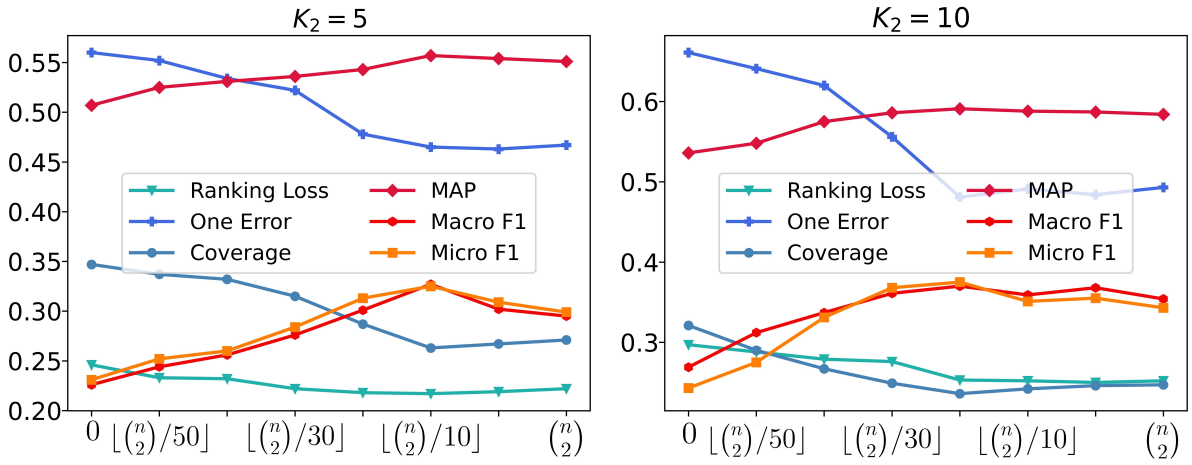


Figure 8: Performance *Ranking Loss*↓, *One Error*↓, *Coverage*↓, *MAP*↑, *MacroF1*↑ and *MicroF1*↑ under different number of generated samples on **NUS-WIDE** ($p = 0.8$ and $r = 1$)

Based on Fig. 8, we can observe the following:
(i) When $K_2 = 5$, MAP, MacroF1 and MicroF1 gradually increase and reach to the maximum when the number of generated samples is about $\lfloor\binom{n_2}{2}/10\rfloor$ and they maintain relatively stable with more generated samples. While Ranking Loss, One Error and Coverage also reach their minimum value when utilizing $\lfloor\binom{n_2}{2}/10\rfloor$ generated samples.
(ii) When K_2=10, FsPML-SF achieves the best performance when the number of generated samples approximates to $\lfloor\binom{n_2}{2}/20\rfloor$ and also maintains a stable performance with more generated samples. This is because more few-shot samples are used for training in the latter case and fewer augmented samples are needed.
These observations again prove the necessity of generating training samples to combat with learning from few-shot samples with noisy labels and suggest the effectiveness of FsPML-SF on generating credible training samples. Based on these results, we set the number of generated samples from $\mathcal{D}_{novel}$ as $\lfloor\binom{n_2}{2}/2K_2\rfloor$.

5. Conclusion

In this paper, we focus on the problem of few-shot multi-label classification using scarce samples with over-annotated labels. Our proposed FsPML-SF conducts label disambiguation to dislodge noisy labels by jointly mining the feature and semantic similarity, label credibility of other samples and label correlations in a principle way, and introduces a synthetic network to extract label-specific features for generating new samples with credible labels. Extensive empirical studies on benchmark datasets demonstrate the advantages of FsPML-SF to competitive methods, and both the label disambiguation and data augmentation improve the performance of FsPML-SF.

Our problem is closely related to the crowdsourced task in practical scenario, where the collected data tends to be over-annotated and with small quantity. Thus, how to adopt our method into the real-world crowdsourced data is the future pursue.

Acknowledgments

We thank the authors who kindly share their datasets or source codes with us for the experimental study, and we also appreciate the anonymous ICDM reviewers for their constructive comments on improving this paper. This work is supported by National Natural Science Foundation of China (No. 62031003, 61872300).

References

[1] M.-K. Xie, S.-J. Huang, Partial multi-label learning, in: AAAI, 2018, pp. 4302–4309.

[2] G. Yu, X. Chen, C. Domeniconi, J. Wang, Z. Li, Z. J. Zhang, X. Wu, Feature-induced partial multi-label learning, in: ICDM, 2018, pp. 1398–1403.

[3] M.-L. Zhang, Z.-H. Zhou, A review on multi-label learning algorithms, TKDE 26 (8) (2013) 1819–1837.

[4] G. Yu, J. Tu, J. Wang, C. Domeniconi, X. Zhang, Active multilabel crowd consensus, TNNLS 32 (4) (2020) 1448–1459.

[5] T. Chen, T. Pu, H. Wu, Y. Xie, L. Lin, Structured semantic transfer for multi-label recognition with partial labels, ArXiv abs/2112.10941.

[6] X. Fang, P. Yan, Multi-organ segmentation over partially labeled datasets with multi-scale feature abstraction, TMI 39 (2020) 3619–3629.

[7] S. Modi, S. Dey, A. Singh, Proportional and derivative controllers for buffering noisy gene expression, in: CDC, 2019, pp. 2832–2837.

[8] J.-P. Fang, M.-L. Zhang, Partial multi-label learning via credible label elicitation, in: AAAI, 2019, pp. 3518–3525.

[9] H. Wang, W. Liu, Y. Zhao, C. Zhang, T. Hu, G. Chen, Discriminative and correlative partial multi-label learning., in: IJCAI, 2019, pp. 3691–3697.

[10] L. Sun, S. Feng, T. Wang, C. Lang, Y. Jin, Partial multi-label learning by low-rank and sparse decomposition, in: AAAI, 2019, pp. 5016–5023.

[11] T. Yu, G. Yu, J. Wang, C. Domeniconi, X. Zhang, Partial multi-label learning using label compression, in: ICDM, 2020, pp. 761–770.

[12] N. Xu, Y.-P. Liu, X. Geng, Partial multi-label learning with label distribution, in: AAAI, 2020, pp. 6510–6517.

[13] G. Lyu, S. Feng, Y. Li, Noisy label tolerance: A new perspective of partial multi-label learning, Inf. Sci. 543 (2021) 454–466.

[14] M. Xie, F. Sun, S. Huang, Partial multi-label learning with meta disambiguation, in: KDD, 2021, pp. 1904–1912.

[15] A. Alfassy, L. Karlinsky, A. Aides, J. Shtok, S. Harary, R. S. Feris, R. Giryes, A. M. Bronstein, Laso: Label-set operations networks for multi-label few-shot learning, CVPR (2019) 6541–6550.

[16] T. Chen, L. Lin, R. Chen, X. Hui, H. Wu, Knowledge-guided multi-label few-shot learning for general image recognition, TPAMI 44 (2022) 1371–1384.

[17] C.-W. Lee, W. Fang, C.-K. Yeh, Y.-C. F. Wang, Multi-label zero-shot learning with structured knowledge graphs, in: CVPR, 2018, pp. 1576–1585.

[18] G. Ou, G. Yu, C. Domeniconi, X. Lu, X. Zhang, Multi-label zero-shot learning with graph convolutional networks, Neural Networks 132 (2020) 333–341.

[19] Y. Zhao, G. Yu, L. Liu, Z. Yan, C. Domeniconi, L. Cui, Few-shot partial multi-label learning, ICDM (2021) 926–935.

[20] J. Snell, K. Swersky, R. S. Zemel, Prototypical networks for few-shot learning, in: NeurIPS, 2017, pp. 4077–4087.

[21] J. Liu, L. Song, Y. Qin, Prototype rectification for few-shot learning, in: ECCV, 2020, pp. 741–756.

[22] Y. Sun, Y. Zhao, G. Yu, Z. Yan, C. Domeniconi, Few-shot partial multi-label learning with data augmentation, ICDM (2022) 478–487.

[23] M.-K. Xie, S.-J. Huang, Partial multi-label learning with noisy label identification, TPAMI 44 (7) (2022) 3676–3687.

[24] T. Cour, B. Sapp, B. Taskar, Learning from partial labels, JMLR 12 (2011) 1501–1536.

[25] M.-L. Zhang, F. Yu, C.-Z. Tang, Disambiguation-free partial label learning, TKDE 29 (10) (2017) 2155–2167.

[26] J. Tu, G. Yu, C. Domeniconi, J. Wang, G. Xiao, M. Guo, Multi-label answer aggregation based on joint matrix factorization, in: ICDM, 2018, pp. 517–526.

[27] T. Yu, G. Yu, J. Wang, M. Guo, Partial multi-label learning with label and feature collaboration, in: DASFAA, 2020, pp. 621–637.

[28] Z. Li, G. Lyu, S. Feng, Partial multi-label learning via multi-subspace representation, in: IJCAI, 2020, pp. 2612–2618.

[29] T. Yu, G. Yu, J. Wang, C. Domeniconi, X. Zhang, Partial multi-label learning using label compression, 2020 IEEE International Conference on Data Mining (ICDM) (2020) 761–770.

[30] G. Lyu, S. Feng, Y. Li, Partial multi-label learning via probabilistic graph matching mechanism, Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.

[31] M.-K. Xie, S.-J. Huang, Semi-supervised partial multi-label learning, 2020 IEEE International Conference on Data Mining (ICDM) (2020) 691–700.

[32] L. Sun, S. Feng, G. Lyu, H. Zhang, G. Dai, Partial multi-label learning with noisy side information, Knowledge and Information Systems 63 (2020) 541–564.

[33] J.-H. Wu, X. Wu, Q. Chen, Y. Hu, M.-L. Zhang, Feature-induced manifold disambiguation for multi-view partial multi-label learning, Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.

[34] X. Gong, D. Yuan, W. Bao, Understanding partial multi-label learning via mutual information, in: NeurIPS, 2021.

[35] L. Sun, S. Feng, J. Liu, G. Lyu, C. Lang, Global-local label correlation for partial multi-label learning, IEEE Transactions on Multimedia 24 (2022) 581–593.

[36] P. Zhao, S. Zhao, X. Zhao, H. Liu, X. Ji, Partial multi-label learning based on sparse asymmetric label correlations, Knowl. Based Syst. 245 (2022) 108601.

[37] X. Gong, D. Yuan, W. Bao, Partial multi-label learning via large margin nearest neighbour embeddings, in: AAAI, 2022.

[38] Y. Yan, Y. Guo, Adversarial partial multi-label learning with label disambiguation, in: AAAI, 2021, pp. 10568–10576.

[39] Y. Yan, S. Li, L. Feng, Partial multi-label learning with mutual teaching, Knowl. Based Syst. 212 (2021) 106624.

[40] A. Rios, R. Kavuluru, Few-shot and zero-shot multi-label learning for structured label spaces, in: EMNLP, 2018, pp. 3132–3142.

[41] Y. Hou, Y. Lai, Y. Wu, W. Che, T. Liu, Few-shot learning for multi-label intent detection, in: AAAI, 2020.

[42] C. Simon, P. Koniusz, M. Harandi, Meta-learning for multi-label few-shot classification, 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (2021) 346–355.

[43] Z. Wang, Y. Duan, L. Liu, D. Tao, Multi-label few-shot learning with semantic inference (student abstract), in: AAAI Conference on Artificial Intelligence, 2021.

[44] L. Xiao, P. Xu, L. Jing, U. Akujuobi, X. Zhang, Semantic guide for semi-supervised few-shot multi-label node classification, Inf. Sci. 591 (2022) 235–250.

[45] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, in: ICLR, 2015.

[46] S. E. Reed, Y. Chen, T. L. Paine, A. van den Oord, S. M. A. Eslami, D. J. Rezende, O. Vinyals, N. de Freitas, Few-shot autoregressive density estimation: Towards learning to learn distributions, in: ICLR, 2018.

[47] J.-Y. Zhu, T. Park, P. Isola, A. A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: ICCV, 2017, pp. 2242–2251.

[48] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, Comm. of ACM 63 (11) (2020) 139–144.

[49] J. Ye, J. He, X. Peng, W. Wu, Y. Qiao, Attention-driven dynamic graph convolutional network for multi-label image recognition, in: ECCV, 2020, pp. 649–665.

[50] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: CVPR, 2016, pp. 2818–2826.

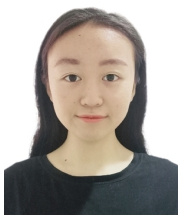
[51] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: ICLR, 2015.

[52] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: ECCV, 2014, pp. 740–755.

[53] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, Y. Zheng, Nus-wide: a real-world web image database from national university of singapore, in: ACM CIVR, 2009, pp. 1–9.

[54] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: CVPR, 2009, pp. 248–255.

Author Biographies

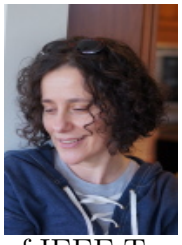


Yifan Sun is a senior student at the School of Software, Shandong University. Her research interests include machine learning and data mining, especially on few-shot learning.

Yunfeng Zhao is a PhD student at the School of Software, Shandong University. He received B.Sc. of Computer Science and Technology from Shandong Normal University in July 2020. His research interests include machine learning and data mining, especially on few-shot learning and federated learning.

Guoxian Yu is a Professor at the School of Software, Shandong University, Jinan, China. He received the Ph.D. in Computer Science from South China University of Technology, Guangzhou, China in 2013. His research interests include data mining and bioinformatics. He has served as Associate Editor for Interdisciplinary Sciences: Computational Life Sciences, BioMed Research International, PC/SPC/AC member for ICML, NeurIPS, IJCAI, AAAI, KDD, and reviewer for many IEEE/ACM Transactions journals.

Zhongmin Yan is an Associate Professor at the School of Software, Shandong University, Jinan, China. She received the B.Sc., M.S. and Ph.D. degrees from Shandong University in 1998, 2001 and 2010, respectively. Her research interests are in the areas of Web information integration and Web data management.



Carlotta Domeniconi is an Associate Professor in the Department of Computer Science at George Mason University. Her research interests include machine learning, pattern recognition, and data mining, with applications in text mining and bioinformatics. She has published extensively in premier journals and conferences in machine learning and data mining. She has served as PC member for KDD, ICDM, SDM, and AAAI. She is an Associate Editor of IEEE Transactions on Knowledge and Data Engineering, and Knowledge and Information Systems.