Generative Correlation Discovery Network for Multi-Label Learning

Lichen Wang*, Zhengming Ding†, Seungju Han‡, Jae-Joon Han‡, Changkyu Choi‡ and Yun Fu*
*Northeastern University, Boston, USA.
†Indiana University-Purdue University Indianapolis, Indianapolis, USA.
‡Samsung Electronics, Advanced Institute of Technology.
wanglichenxj@gmail.com, zd2@iu.edu, {sj75.han, jae-joon.han, changkyu_choi}@samsung.com, yunfu@ece.neu.edu

***Abstract*—The goal of Multi-label learning is to predict multiple labels of each single instance. This is a challenging problem since the training data is limited, long-tail label distribution, and complicated label correlations. Generally, more training samples and label correlation knowledge would benefit the learning performance. However, it is difficult to obtain large-scale well-labeled datasets, and building such a label correlation map requires sophisticated semantic knowledge. To this end, we propose an end-to-end Generative Correlation Discovery Network (GCDN) method for multi-label learning in this paper. GCDN captures the existing data distribution, and synthesizes diverse data to enlarge the diversity of the training features; meanwhile, it also learns the label correlations based on a specifically-designed, simple but effective correlation discovery network to automatically discover the label correlations and considerately improve the label prediction accuracy. Extensive experiments on several benchmarks are provided to demonstrate the effectiveness, efficiency, and high accuracy of our approach[1].**

I. Introduction

Single-label learning assumes that each instance only belongs to a single label/category. However, with exponentially growing data, a lot of real-world data mining tasks are required to assign more than one labels to a single instance. For example, real-world objects are various and sophisticated, thus, one object can be annotated by tens or hundreds of descriptions such as color, shape, texture and category. As a consequence, multi-label learning emerged to handle such challenges [1]. Formally, multi-label learning searches a mapping from the original feature space to the label space. It has become an attractive research area in recent years due to its vast potential in real-world applications such as image annotation [2], [3], [4], large-scale image retrieval [2], [3], semantic analysis [5], data mining [6] and recommendation systems [7].

There are two major and unique challenges in multi-label learning scenario. *First*, the training sets cannot cover the entire test/real feature space due to the small scale of the existing datasets. As shown in Figure 1, if test samples fall into the light color (training-testing non-overlap) region, the performance would decrease significantly. Moreover, most labels follow a long-tail distribution, which means some labels rarely show up (e.g., *Research*) while others are much common

[1]Code is provided on:https://github.com/wanglichenxj/Generative-Correlation-Discovery-Network-for-Multi-Label-Learning

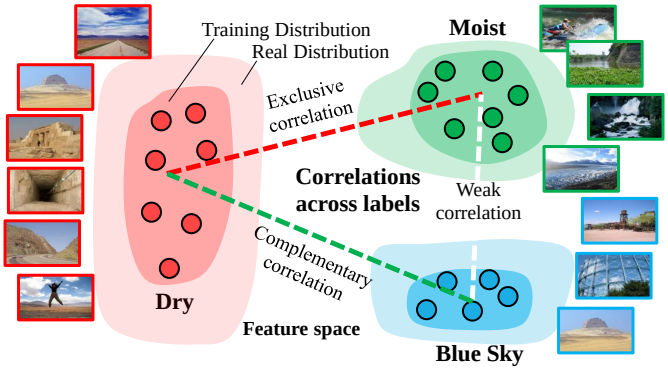

Fig. 1. Two unique challenges exist in multi-label learning scenario. 1) The training data distribution (dark color region) is usually smaller than testing/real data distribution (light color region) due to the limited data scale and long-tail feature distribution. 2) The label correlations are sophisticated and crucial for multi-label prediction. For instance, ***Dry*** and ***Moist*** are usually exclusive, while ***Dry*** and ***Blue Sky*** have higher possibility to appear together, and the correlation between ***Blue Sky*** and ***Moist*** is weak.

(e.g., *Natural Light*) [8]. More training samples could solve this problem. However, building such dataset is much more expensive compared with single-label dataset. Labeling errors usually occur in multi-label learning scenario since some labels are subjectively assigned (e.g., *stressful* and *congregating*), while different people hold various opinions. *Second*, the label correlations are crucial to make accurate prediction [9], [10], [11]. As illustrated in Figure 1, *Dry* and *Moist* cannot exist simultaneously; while *Dry* and *Blue Sky* sometimes show up together; and there is no significant correlation between *Blue Sky* and *Moist*. Involving this knowledge could improve the performance significantly, however, most existing datasets/applications do not have this information. Besides, building such correlation map needs specialized semantic knowledge, and the built map cannot generalize well to other multi-label datasets which obviously limits the deployment to a wide range of real-world multi-label learning applications.

These issues are common in most multi-label learning datasets. For instance, ESP dataset [12] has $18,689$ samples for training, however, there are only 4.69 labels in average assigned to each sample from 268 candidate labels. Some conventional multi-label learning methods deploy sophisticated constraints to avoid over-fitting in supervised scenario. Others utilize unlabeled auxiliary data to explore more label

information in semi-supervised scenario [13]; However, it is easy to cause negative transfer if the distributions of source and target domains are significantly different.

In this work, we explore the idea of the Generative Adversarial Network (GAN) [14] and propose a novel Generative Correlation Discovery Network (GCDN, shown in Figure 2) for multi-label learning. In summary, GCDN captures the feature distribution of each label, and generates fake features, which completes the distribution to obtain more general samples. Meanwhile, GCDN learns the correlations across different labels and takes advantage of the learned semantic structure knowledge to significantly improve the learning performance. Our main contributions are listed as follows:

- A specifically-designed multi-label conditional feature generative strategy is proposed. It synthesizes and diversifies the feature space to improve the model robustness and generalization.
- A graph-based Correlation Discovery Network (CDN) is proposed to automatically learn semantic correlations across different labels and utilize the knowledge to further improve learning performance.
- A similarity constraint is deployed associated with the multi-label prediction to stabilize the generator training, which is effective in multi-label learning scenario.

All designed networks are trained simultaneously in an end-to-end scenario without other semantic information as prior knowledge, which is easy to deploy to a wide range of potential multi-label learning and relevant applications.

II. Related Work

A. Multi-label Learning

Multi-label learning searches for patterns from the instances that are associated with a set of labels. Problems with Multi-label learning are common in a wide range of real-world applications, such as image classification [1], [15], text classification [16], [17] as well as video concept recognition [18]. Compared with the single-label scenario, the multi-label scenario is more challenging [1]. For instance, if there is a large number of label candidates, the task would become difficult since the number of possible label set will become tremendous. And the combinations across various labels.

Multi-label learning can be separated into two settings: supervised and semi-supervised [1], [19], [13], [20] scenarios. Supervised approaches need a large amount of labeled training samples to reach high performance. [21] proposed an efficient approach to eliminate the label noise. MEFF [22] utilizes a multi-view fusion approach for multi-label classification. Modulation approach is proposed in [23] for encouraging the coupling of relevant tasks for image retrieval. However, the scales of multi-label datasets [24], [12], [25], [26] are always limited which reduces the potential of the approaches. Semi-supervised learning [27], [28] is able to explore a more compatible model by making use of a small scale labeled dataset as well as a large scale unlabeled data [19], [13]. However, the performances of these kinds of approaches are significantly rely on the quality of the auxiliary data and the optimization process is complicated which is hard to control [29], [30]. Moreover, in multi-label scenario, label correlation is crucial and important to further improve the learning performance. [31], [10], [32] builds a semantic label hierarchy as prior knowledge to generate a label dependency graph. [9] utilizes an label semantic structure to deduce label noise and cover diverse and distinct labels. Label embedding [33] projects labels into a latent space to explore the label relations. [34] uses attention and RNN based approach to obtain the object relations in an image space. However, these approaches require the pre-defined label correlation information which is expensive and difficult to obtain.

We proposed a generative model which belongs to supervised learning scenario. It automatically learns the feature distribution from the training data and even the visual components across different samples. Our approach is able to span the feature area and overcome the limited training data scenario. In addition, inspired by the use of graph strategy in deep model [35], such as Graph-based CNN [36]; in our work, we design a simple but effective Correlation Discovery Network (CDN) to learn the correlation among different labels.

B. Generative Adversarial Net (GAN)

GAN [14] contains two neural network structures: First, a generator network is trained to generate fake samples and confuse a discriminator network. Second, the discriminator tries to differentiate the real and generated samples. The generator and the discriminator which are trained in opposition to one another. The competition between the two networks lets both of them to enhance their abilities until the fake samples are indistinguishable. Many variations of GAN are proposed for various goals and applications. Least Squares Generative Network [37] adopts the least squares loss objective for the discriminator. It overcomes the vanishing gradient challenges during the training process. Mode Regularized GAN [38] introduces several ways of regularizing the objective function, which can dramatically stabilize the training of GAN models. Cycle GAN [39] proposes a structure which translates utilizes the absence of paired examples for source and target domain translations. GMVAR [40] generates samples of different views for multi-view classification task. Conditional GAN (CGAN) [41], [42] extends GAN strategy by adding conditional knowledge, such as classification labels, on both the discriminator and generator. ACGAN [43], [44] is proposed based on CGAN but is specifically associated with an auxiliary classifier, which is utilized to guide and stabilize the training process for the generator. However, CGAN and ACGAN were mainly designed to subjectively diversify images and utilize the human perceptual model MS-SSIM [45] to evaluate the generation diversity. It is not designed for objective classification purposes. Conditional Loss-Sensitive GAN [46] designs a loss function to make the fake images more real and can also classify target images. However, it is designed for single label classification and is difficult to extend to the multi-label scenario since it utilizes optimization strategy to classify.

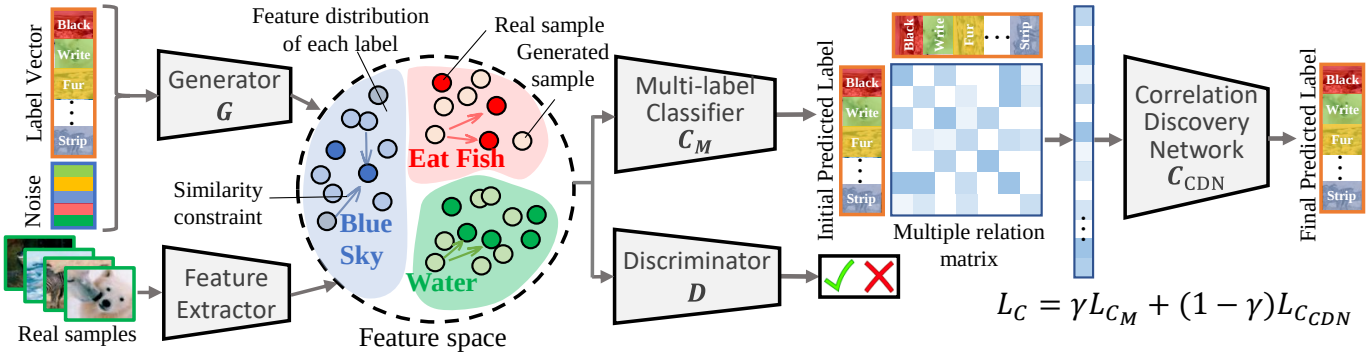

Fig. 2. Framework of our approach, where a generator $G(\cdot)$, a discriminator $D(\cdot)$, and a multi-label classifier $C_M(\cdot)$ are simultaneously trained. The generator synthesizes augmented samples conditioned on the provided labels to handle the limited data and long-tail label distribution drawbacks; while the classifier predicts initial multi-label results, and the results are transferred to correlation discovery network to learn correlations and obtain final high accuracy results. All networks are jointly trained in an end-to-end scenario to achieve the highest performance.

Different from previous works, our approach is proposed to explore the generative model in the multi-label learning scenario. Specifically, our model applies to multi-label classification rather than single-label classification. The model builds connections across labels and features, and is designed to increase the visual feature diversity for boosting learning performance rather than increasing diversity for the subjective human perceptual evaluation [45].

III. The Proposed Approach

A. Preliminaries & Motivation

Given the multi-label training data $\{X_l, Y_l\}$, $X_l \in \mathbb{R}^{d \times n_l}$ is the feature matrix, where each column $x_i \in \mathbb{R}^d$ represents one instance, n_l is the instance number, and d is the feature dimension. $Y_l \in \mathbb{R}^{d_l \times n_l}$ is the label matrix, where d_l is the dimension of the label. Each column y_i denotes the corresponding multi-label vector of x_i. Generally, our approach aims to train based on $\{X_l, Y_l\}$ without any other prior knowledge, and predict the multi-label Y_u of X_u. Since the feature space is much more diverse than the label space, thus, it is challenging to collect enough labeled visual data to capture the data variance. Moreover, there are sophisticated correlations residing across different labels. It is useful and crucial information to further improve the learning performance, but it is difficult and expensive to obtain.

To this end, we aim to compensate the visual feature and mitigate the gap between the training and testing samples. Inspired by the idea of generative model, it is natural to synthesize more diverse features conditioned on each multi-label vector. Meanwhile, a simple but effective graph structure is proposed to automatically explore the label correlation knowledge to further improve the learning performance. These two parts are crucial to improve the learning performance, since it allows the model to fully utilize the feature-label mapping and label-label correlation knowledge from both the feature space and label space of the training samples.

B. Our Approach

Figure 2 illustrates the structure of our approach. It contains a generator $G(\cdot)$, a discriminator $D(\cdot)$, a multi-label classifier $C_M(\cdot)$, and a correlation discovery network $C_{CDN}(\cdot)$. $X_g = G(z|Y)$, where Y is the label matrix for conditionally generating samples and z is the random noise. The $D(\cdot)$ outputs the probability of the samples being real or fake. The generator captures the feature distribution of the existing data and borrow shared components from other categories. $C_{CDN}(\cdot)$ further learns the label correlation and helps to improve the final label prediction. The objective function of $D(\cdot)$ is shown in Eq. (1) which manages to maximize L_D:

$$\begin{aligned} L_D = \; & E_{X \sim p_X(X)} \log D(X|Y) \\ & + E_{z \sim p_z(z)} \log(1 - D(G(z|Y))), \end{aligned} \tag{1}$$

where $D(\cdot)$ is a three-layer network including a fully connected layer with ReLU activation, a mini-batch [47] layer with the LeakyReLU [48] activation, and a fully-connected layer with Sigmoid function. Multi-label classifier $C_M(\cdot)$ includes two objectives. The first one is trained based on real samples, while the second one is based on the generated samples associated with the conditional labels to improve the robustness and generalization of the classifier. The objective function is shown as follow:

$$\begin{aligned} L_{C_M} = \; & \mu \|Y - C_M(X)\|_{\mathrm{F}}^2 \\ & + (1-\mu) \|Y - C_M(G(z|Y))\|_{\mathrm{F}}^2, \end{aligned} \tag{2}$$

where μ is the trade-off parameter which is used to balance the weights between real and fake samples. μ is empirically set to 0.5 in our implementation, which expects both the real and fake samples are evenly utilized for training. And it also avoids extra parameter tuning. Meanwhile, we have observed that slightly tuning μ near 0.5 does increase the performance a little, and cross validation could be employed for automatic parameter tuning. $C_M(\cdot)$ is a two-layer network with ReLU activation in the hidden layer and a Sigmoid in the output layer. We observe that two layers are enough for label prediction, and the model is not sensitive to the number of layers. For

discriminator, we include more constraints. The first term is the major competing component with $D(\cdot)$ and makes the generated samples as real as possible:

$$L_{Gd} = -E_{z\sim p_z(z)} \log(1 - D(G(z|Y))). \tag{3}$$

Compared with single label learning, multi-label learning provides more abstract information for each sample. Inspired by ACGAN [43], we further utilize the classification results as another clue to stabilize the generator training:

$$L_{Gc} = \|Y - C_M(G(z|Y))\|_{\mathrm{F}}^2. \tag{4}$$

Moreover, considering the various feature distributions across labels, the proposed terms may not be strong enough to achieve stable and robust generation performance. Thus, we further include similarity constraint which pulls the generated samples and real samples to be similar:

$$L_{Gs} = \|G(z|Y) - X\|_{\mathrm{F}}^2. \tag{5}$$

After combining all the objectives together and the generator loss is shown as follow:

$$L_G = L_{Gd} + \alpha L_{Gc} + \lambda L_{Gs}, \tag{6}$$

where α and λ are the trade-off parameters which balance the scales across binary discriminator loss, multi-label space, and visual feature space. The major goal of L_{G_s} is to stabilize the training process, and we could tune λ to balance the strength of L_{G_s}. We did observe that large λ decreased the final performance, while a small-scale λ on L_{G_s} indeed reduced training fluctuation and sped up the training process. $G(\cdot)$ is a two-layer neural network in addition with a batch-normalization layer [49] to normalize input vector and improve model robustness.

Simply deploying GAN model is not enough to achieve the highest performance. As introduced before, label correlation is crucial to further improve learning performance. Thus, we propose a simple while effective Correlation Discovery Network (CDN), $C_{CDN}(\cdot)$, to automatically explore the label correlation knowledge (Figure 2). After the predicted label $f_{ci} = C_M(x_i)$ is obtained, where $f_{ci} \in \mathbb{R}^{d_l \times 1}$ is the prediction of each instance x_i. We make a transformation from f_{ci} to an adjacency matrix $\mathbf{m_{ci}}$ by multiplying f_{ci} and its transposition as $\mathbf{m_{ci}} = f_{ci} \times f_{ci}^{\top}$, where $\mathbf{m_{ci}} \in \mathbb{R}^{d_l \times d_l}$ is the adjacency matrix and d_l is the label dimension. The obtained $\mathbf{m_{ci}}$ is reshaped to a $\mathbb{R}^{d_l^2 \times 1}$ vector and forwarded to a fully connected layer network and further predicts the multi-label result. To this end, the objective of CDN is shown below:

$$L_{C_{CDN}} = \sum_{i=1}^{n_l} \|y_i - C_{CDN}(C_M(x_i)C_M(x_i)^{\top})\|_2^2, \tag{7}$$

where $y_i \in \mathbb{R}^{d_l \times 1}$ is the corresponding multi-label vector of x_i. In this network, the elements in $\mathbf{m_{ci}}$ are the multiplication of each pair of the predicted labels of f_{ci}, which could be considered as a similarity metric of the pairwise labels (including the similarity with itself). CDN is trained based on the similarities structure. By this way, CDN explores the latent correlation knowledge residing inside the training data based on the obtained similarities, and further refines the predicted label from $C_M(.)$ to improve performance.

In summary, $C_M(\cdot)$ obtains initial (low-accurate) results first, then $C_{CDN}(\cdot)$ further utilizes the available prediction to "tune" the result to high-accurate. Specifically, $C_{CDN}(.)$ can be considered as a refine strategy over $C_M(.)$. It explores the latent structure knowledge (correlation) across labels and further improves the prediction performance. Jointly optimizing $C_M(.)$ and $C_{CDN}(.)$ by combining their losses together could 1) control the training of $C_M(.)$ to predict rough labels and 2) intentionally force $C_{CDN}(.)$ to capture the label correlations based on the rough labels from $C_M(.)$. This strategy balances the update processing between $C_M(.)$ and C_{CDN} to further help each other in the training stage and achieve the promising performance at last. To this end, the objective function is shown as below:

$$L_C = \gamma L_{C_M} + (1 - \gamma) L_{C_{CDN}}, \tag{8}$$

where $\gamma \in [0, 1]$ is the trade-off parameter which is used to balance the weights of the two objective terms. We empirically set $\gamma = 0.5$ for the experiments, and its parameter sensitivity will be analyzed in the following sections.

In our implementation, $C_{CDN}(\cdot)$ is a fully-connected two-layer network with ReLU activation in the first layer and Sigmoid activation before output. Considering $\mathbf{m_{ci}}$ is a symmetric matrix, thus, to reduce the redundant weights, we remove almost half of the $\mathbf{m_{ci}}$ and forward to $C_{CDN}(\cdot)$. This strategy improves model efficiency without losing any information, and the input dimension of $C_{CDN}(\cdot)$ becomes to $(d_l^2 + d_l)/2$.

C. Discussion

Our proposed model contains three networks jointly optimized in a minimax strategy, which brings in several advantages. First, it is an end-to-end framework without the requirement of any other prior knowledge (e.g., semantic label hierarchy), which is easy to train and compatible for a wide-range of applications; second, the learning performance is robust. Since the generated data enlarges and diversifies the feature distribution, which effectively reduces the overfitting issue; third, other than the discriminator, the classifier as well as the similarity constraint further guide the generator optimization process and make the training process be efficient and stable; fourth, the GCDN can be directly deployed for testing without any other optimization operations which is more simple and efficient in inference compared with graph-based approaches. To this end, our model jointly trains the components and enables each component to benefit others. The experiments demonstrate its necessity in multi-label scenario.

Compared with the conventional generative model, our approach is different in the following aspects. First, our model conditions multi-label information (either binary or continuous values) to synthesize the visual features which is more challenging than the single-label generation scenario. Second, the label correlation knowledge is automatically learned in the training procedure without any extra semantic knowledge,

which is more compatible in a wide range of real-world application. Third, our model generates samples in feature space instead of image space, thus, it is not only limited for image level application, but also potentially works well with other data types (which is demonstrated in experiments).

IV. Experiment

We evaluate our approach associated with the state-of-the-art approaches on six fine-grained datasets. We further extend the experiments to zero-shot multi-label learning, image annotation, as well as image retrieval scenarios in the experiments.

A. Multi-label Datasets

Six image datasets are utilized for evaluation. Brief introductions of the datasets are as follows:

Corel5K Dataset [24] is a subset from the Corel Photo CD dataset. It contains $4,500$ images assigned for training and 499 images assigned for testing. Each label is a 260-dimensional semantic description vector in binary format. The average descriptions per sample is 3.40.

ESP Game Dataset [12] is labeled by an ESP interactive system, which is designed like a computer game in the labeling process. It includes $18,689$ samples assigned for training and $2,081$ samples assigned for testing. The label vector is a 268-dimensional vector in binary value. On average, each sample is assigned with 4.69 labels.

IAPRTC-12 Dataset [50] CLEF cross-language dataset which is generated for image retrieval task. It has $19,627$ samples including landscapes, animals, actions, etc. $17,665$ samples are assigned for training and $1,962$ samples are assigned for testing. The label vector is 291-dimension in binary format with averagely 5.72 labels.

SUN Dataset [26] is a scene multi-label database including images such as *bakery*, *ballroom*, and *balcony*. There are 717 scene classes in total. Each instance contains a 102-dimensional label vector in a continuous value format, ranged between $[0, 1]$ assigned by multiple trained labors, with averagely 6.31 labels per sample. There are $12,900$ samples for training and $1,440$ samples for testing.

CUB Dataset [25] has 200 birds. Each instance has roughly 31.39 annotations in a binary 312-dimensional label vector. There are several options to split the images for training and testing with roughly $8,800$ samples for training and $1,440$ samples for evaluation.

AWA Dataset [51] consists of more than $30,000$ images captured from 50 animal species. The label vector is a 85-dimensional vector and each instance has roughly 15 labels. Different from other datasets, the label vectors are continuous values that range from 0 to 100. There are $24,295$ samples for training and $6,180$ samples for testing.

B. Experimental Setup

In our implementation, all three networks are fully connected networks. Other sophisticated deep networks can also be applied to attain higher performance.

For ESP Game, IAPRTC, and Corel5K datasets, we utilize 15 different visual descriptors, which are extracted by [52].

TABLE I
MULTI-LABEL LEARNING PERFORMANCE

Data	Method	Pre	Rec	F1	N-R	mAP
Corel	LR	0.2859	0.3211	0.3025	128	0.3630
	SSMLDR	0.2741	0.3366	0.3022	143	0.3410
	FastTag	0.3123	0.3657	0.3369	143	0.3871
	ML-PGD	0.2575	0.2911	0.2732	122	0.3727
	SAE	0.2962	0.3442	0.3184	141	0.3823
	AG2E	0.3011	0.3520	0.3245	**157**	0.3568
	Ours	**0.3335**	**0.3714**	**0.3514**	148	**0.4417**
ESP	LR	0.3793	0.2038	0.2653	215	0.3440
	SSMLDR	0.3298	0.1885	0.2399	226	0.3156
	FastTag	0.4011	0.1927	0.2617	208	0.3904
	ML-PGD	0.3239	0.2012	0.2482	210	0.4077
	SAE	0.3861	0.1743	0.2402	194	0.3842
	AG2E	0.3548	0.1525	0.2133	213	0.3730
	Ours	**0.4032**	**0.2178**	**0.2828**	**239**	**0.4327**
IAP	LR	0.4287	0.2041	0.2765	199	0.4211
	SSMLDR	0.3491	0.2520	0.2927	229	0.3981
	FastTag	0.4346	0.2267	0.2980	227	0.4596
	ML-PGD	0.4132	0.2441	0.3011	230	0.4674
	SAE	0.3537	0.2282	0.2774	213	0.4309
	AG2E	0.3829	0.2330	0.2897	229	0.4353
	Ours	**0.4732**	**0.2648**	**0.3396**	**237**	**0.5295**
SUN	LR	0.6209	0.1473	0.2457	102	0.6807
	SSMLDR	0.6879	0.1700	0.2726	102	0.6723
	FastTag	0.6816	0.1473	0.2457	102	0.6914
	ML-PGD	0.7110	0.1614	0.2631	101	0.7087
	SAE	0.7183	0.1638	0.2668	98	0.7012
	AG2E	0.7685	0.1765	0.2871	99	0.6778
	Ours	**0.7985**	**0.1835**	**0.2985**	**102**	**0.7093**
CUB	LR	0.2010	0.0239	0.0428	157	0.0638
	SSMLDR	0.3410	0.0473	0.0832	178	0.2329
	FastTag	0.2147	0.0359	0.0615	167	0.3144
	ML-PGD	0.3334	0.0451	0.0794	155	0.3288
	SAE	0.3383	0.0514	0.0908	196	0.3255
	AG2E	0.3409	0.0531	0.0911	190	0.3106
	Ours	**0.3718**	**0.0541**	**0.0944**	**214**	**0.3561**
AWA	LR	0.8798	0.0821	0.1500	75	0.8626
	SSMLDR	0.7812	0.0858	0.1546	67	0.8346
	FastTag	0.7861	0.0949	0.1694	72	0.8791
	ML-PGD	0.5395	0.0635	0.1136	57	0.9121
	SAE	0.9683	**0.0957**	**0.1742**	73	**0.9397**
	AG2E	0.8483	0.0827	0.1507	73	0.9033
	Ours	**0.9716**	0.0871	0.1599	**83**	0.9291

For AWA, CUB, and SUN dataset, due to the limited training data which is difficult obtain a well trained convolutional neural network from scratch; hence, the pre-trained VGG Networks [53] based on ImageNet [54] is deployed to extract deep visual features. As shown in Figure 2, the label vector concatenated with random noise is set as input to $G(\cdot)$. α is empirically set to 0.01. λ limits the feature scales which is set to 5 for VGG [53] features and 20 for handcrafted features [52]. ADAM optimizer [55] is employed and the learning rates are set to 0.00002, 0.00002, 0.00005, and 0.001 for $C_M(\cdot)$ and $C_{CDN}(\cdot)$, $D(\cdot)$, and $G(\cdot)$, respectively. In the training procedure, $C_M(\cdot)$ and $G(\cdot)$ are pre-trained to have stable initialization, while $G(\cdot)$ is optimized by $L_G = L_{Gc} + \frac{\lambda}{\alpha} L_{Gs}$ without including L_{Gd} at first, and after around 50 epoch, we switch L_G back to Eq. (7) and train $D(\cdot)$ simultaneously with the other networks. The same number

TABLE II
MULTI-LABEL LEARNING PERFORMANCE ON AUGMENTED LABEL SETS

Data	Methods	Pre	Rec	F1	N-R	mAP
Corel-A	LR	0.2842	0.2304	0.2545	103	0.3762
	SSMLDR	0.3036	0.2791	0.2908	134	0.3660
	FastTag	0.3329	0.3145	0.3234	136	0.4127
	ML-PGD	0.3245	0.3011	0.3124	140	0.4275
	SAE	0.3168	0.3037	0.3101	128	0.4192
	AG2E	0.3273	0.3172	0.3221	**143**	0.3985
	Ours	**0.3438**	**0.3219**	**0.3325**	138	**0.4773**
ESP-A	LR	0.3848	0.1256	0.1894	178	0.3913
	SSMLDR	0.3253	0.1697	0.2231	202	0.3357
	FastTag	0.3886	0.1531	0.2197	196	0.4254
	ML-PGD	0.3713	0.1184	0.1795	162	0.4211
	SAE	0.3153	0.1425	0.1966	156	0.4050
	AG2E	0.3518	0.1492	0.2095	196	0.4030
	Ours	**0.4772**	**0.1944**	**0.2763**	**225**	**0.4436**

of generated and real samples are utilized in each training iteration. We randomly separate the samples into a training and a testing subset with relatively even sample numbers and run our model 5 times and report the average performance. The model is implemented on TensorFlow and trained with Nvidia Titan XP GPU for acceleration. The regular training time is around 20 minutes for model convergence.

C. Multi-label Classification

For the multi-label classification scenario, we evaluate our approach on two settings: (a) Conventional multi-label learning; (b) Zero-shot multi-label learning which is a more challenging task. We compare our approach with several state-of-the-art representative multi-label learning approaches. Brief introductions of the methods are listed below:

- **Least Square Regression (LR)** is a straightforward linear regression model, which learns a linear mapping between the feature and label spaces.
- **Semi-Supervised Multi-Label Dimension Reduction (SSMLDR)** [56] effectively utilizes the information from both labeled and unlabeled data by designing a special label propagation strategy to improve the model's robustness and accuracy.
- **Fast Image Tagging (FastTag)** [21] proposes two co-regularized linear mappings in one loss function. It is able to infer the full list of tags based on the incomplete ground truth training labels.
- **Multi-Label learning using a Mixed Graph (ML-PGD)** [31] proposes a label dependencies model by constructing a mixed graph and combines instance level similarity with class co-occurrence.
- **Semantic AutoEncoder (SAE)** [57] proposes an effective auto-encoder with an additional reconstruction constraint to recover labels.
- **Adaptive Graph Guided Embedding (AG2E)** [58] proposes an adaptive graph strategy which jointly obtains the similarity graph and predicts multiple label in a semi-supervised fashion.

TABLE III
ZERO-SHOT MULTI-LABEL LEARNING PERFORMANCE

Data	Method	Pre	Rec	F1	N-R	mAP
SUN	LR	0.7047	0.1548	0.2539	97	0.6616
	SSMLDR	0.6637	0.1481	0.2422	95	0.6581
	FastTag	0.6906	0.1522	0.2494	90	0.6706
	ML-PGD	0.7037	0.1471	0.2433	95	0.6829
	SAE	0.6978	0.1710	0.2747	100	0.6513
	AG2E	0.7125	0.1618	0.2637	88	0.6693
	Ours	**0.7531**	**0.1857**	**0.2979**	**101**	**0.6911**
CUB	LR	0.2600	0.0307	0.0549	160	0.2693
	SSMLDR	0.2926	0.0383	0.0677	166	0.2329
	FastTag	0.2231	0.0434	0.0726	143	0.2967
	ML-PGD	0.2392	0.0365	0.0635	117	0.3178
	SAE	0.2552	0.0469	0.0798	167	0.3102
	AG2E	0.2808	0.0481	0.0821	163	0.2693
	Ours	**0.3091**	**0.0488**	**0.0843**	**179**	**0.3264**
AWA	LR	0.7555	0.0766	0.1392	66	0.8809
	SSMLDR	0.7017	0.0764	0.1378	66	0.7858
	FastTag	0.8610	0.0912	0.1649	81	0.8918
	ML-PGD	0.4338	0.0623	0.1091	49	0.8677
	SAE	0.9015	**0.0926**	**0.1679**	78	**0.8918**
	AG2E	0.8247	0.0811	0.1476	71	0.8874
	Ours	**0.9249**	0.0804	0.1480	**83**	0.8784

For the SSMLDR method, we directly set testing data as unlabeled data and evaluate its recovery performance. To fully compare our approach with other methods, we utilize the same metrics adopted in [52]. When the labels are recovered, we select the top 5 ranked labels as the recovered label. Then, the recovery precision (Pre) P and the recall (Rec) R are calculated. $P = \frac{t_p}{t_p+f_p}$, and $R = \frac{t_p}{t_p+f_n}$, where t_p represents truth-positive. f_p and f_n represent the false positive and the false negative respectively. To compare the results easier, we calculate the F1-score (F1) which is the harmonic mean of the precision and the recall, where $F1 = 2\frac{P \times R}{P+R}$. We further obtain the number of labels with a non-zero recall (N-R) value. The mean average precision (mAP) from [31] is utilized for comprehensive evaluation. In all metrics, higher value indicates better performance.

The experimental result in conventional multi-label learning setting is illustrated in Table I. We can see that our approach significantly outperforms other baselines in most of the metrics, which demonstrates the high accuracy and robustness of our approach. The work of [31] proposes a complete/augmented label set for Corel5K and ESP Game datasets, increasing Corel5K label from averagely 3.40 to 4.84 labels, and the ESP Game label from 4.69 to 7.27 labels. We evaluate our model based on these label sets (II). The results are shown in Table II, and it indicates that our approach still achieves the best performance in most matrices.

D. Zero-shot Multi-label Classification

We extend our method to the zero-shot multi-label scenario where the classes in training and test are non-overlapped. It is a more challenging task since the distribution gaps are more significant.

We evaluate our model based on SUN, CUB and AWA datasets. These datasets have default training and testing splits

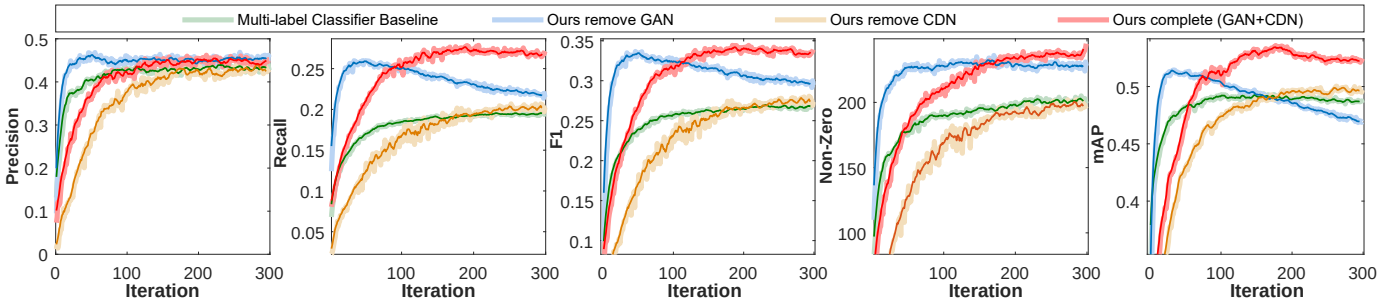

Fig. 3. Ablation study: classification performance along training iterations in the IAPRTC-12 dataset. Different color indicates generative and CDN modules are removed/deployed in our approach. The red line indicates the results of our complete approach; blue line is our model without generative strategy; yellow line is our model without CDN; and green line is the result which both the generative and CDN modules are removed. It illustrates that CDN dramatically improves the learning performance in all metrics especially Recall, F1, and mAP metrics. Only CDN-based strategy causes over-fitting easily due to the limited training data and long-tail feature distribution, while generative model could effectively increase the robustness and stabilize the learning performance. The result demonstrates the effectiveness of both generative and CDN modules in our approach. (Please view the color figures for better visualization)

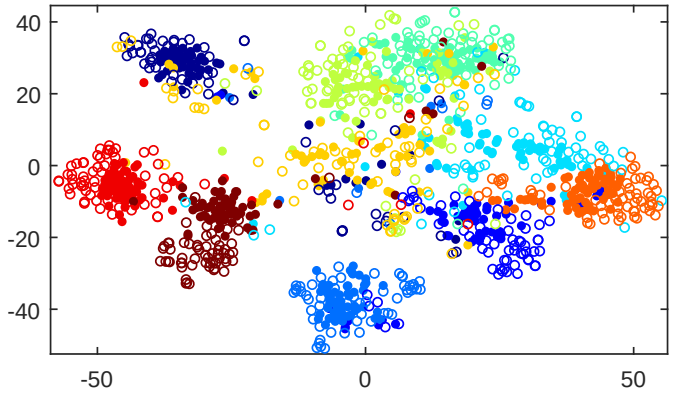

Fig. 4. Visualization of 10 hard unseen classes of both generated (hollow circle) and ground-truth (solid circle) samples. The same color denotes the same class samples. It further demonstrates the generated samples are simimlar but not same to groundtruth samples, and they do enlarge/diversifie the distribution area.

for ZSL. In SUN dataset, 645 classes are used for training and 72 classes are used for testing. In AWA dataset, 40 classes of animals are assigned for training and the other 10 are set for testing. In CUB dataset, 150 classes are set for training and 50 are set for testing. There are 4 different splits in CUB, thus, we execute the testing four times and report the mean results. For SUN and AWA datasets, we run the testing five times and obtain the average performance.

The performance is illustrated in Table III, which indicates the high performance of our approach compared with other baselines. It shows that our model is robust and works well even when the testing classes are unobserved during the training stage. This advantage is suitable for real-world applications since the target images are not controllable. In AWA dataset, our model still cannot achieve the best performance of all metrics. The reasons are similar to the explanation discussed in the conventional multi-label scenario. Since the label distribution is narrow and the scale of training samples is large, the performance of our model cannot achieve significant improvement.

Moreover, we visualize 10 unseen classes of CUB dataset

TABLE IV
MULTI-LABEL LEARNING PERFORMANCE OF ADDING VARIOUS LOW-LEVEL GAUSSIAN NOISES TO THE ORIGINAL FEATURE OF CUB DATASET.

Noise	Pre	Rec	F-1	N-R	mAP
0.00	**0.3718**	**0.0541**	**0.0944**	**214**	**0.3561**
0.05	0.3711	0.0540	0.0941	214	0.3561
0.10	0.3692	0.0538	0.0943	214	0.3537
0.15	0.3668	0.0537	0.0941	214	0.3511
0.20	0.3647	0.0534	0.0938	212	0.3482
0.25	0.3612	0.0533	0.0936	211	0.3467
0.30	0.3591	0.0531	0.0932	209	0.3416
0.35	0.3505	0.0530	0.0930	208	0.3389
0.40	0.3393	0.0529	0.0929	206	0.3351
0.45	0.3314	0.0528	0.0927	204	0.3232
0.50	0.3248	0.0526	0.0926	202	0.3215

from the predicted labels. Specifically, we deploy t-SNE [59] to map both the generated samples (hollow circle) and the ground truth samples (solid circle) to a 2-D subspace in Figure 4. We can see that the samples belong to the same class become closer, while different classes samples are more separated. It indicates that our model could improve discriminability and generalizability to predict multiple labels given an unseen image.

E. Ablation Study

To demonstrate the effectiveness of all the proposed strategies in our approach, we intentionally run our approach with or without the generative and CDN modules in IAPRTC-12 dataset. Figure 3 illustrates the performance with the iteration increasing; different color indicates different settings and details are introduced in the caption. We can see that our approach achieves the highest performance when both generation and CDN modules are deployed. CDN improves the learning performance significantly; however, only utilizing CDN without generative strategy could easily cause over-fitting due to the long-tail label distribution, while generative strategy diversifies the feature distribution and effectively reduces the over-fitting issue. From Figure 3, we observe that only generative model without CDN could also improve the

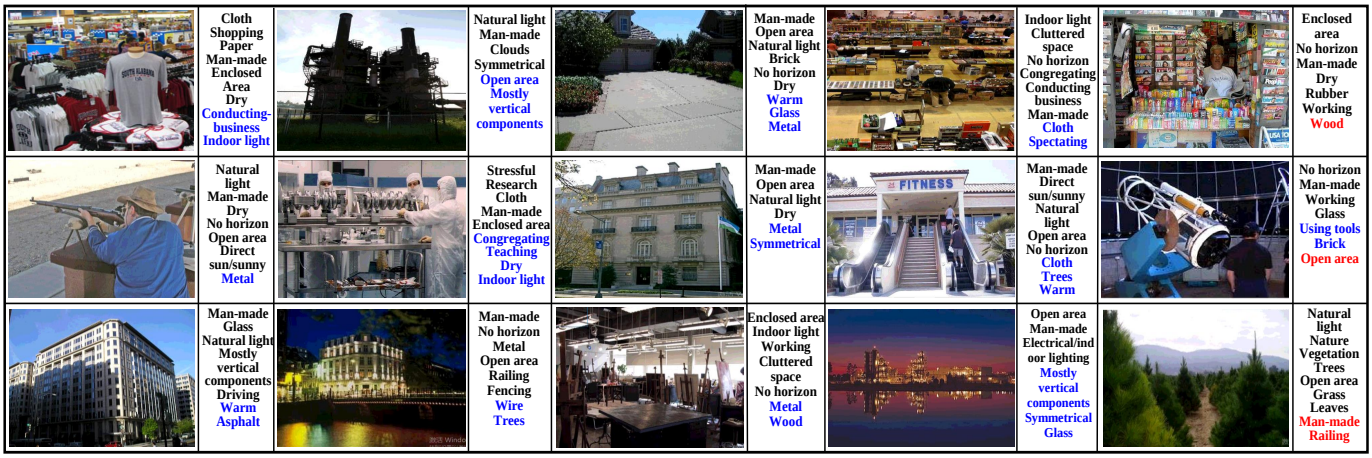

Fig. 5. Samples of recovered labels from SUN dataset. Each image contains several semantic labels. **Black** font denotes labels that match with the ground truth. **Blue** font denotes labels that do not exist in the ground truth but match our judgments. **Red** font denotes incorrect labels from our model. The result shows that our approach is robust and able to recover labels even when labels are missed from the ground truth.

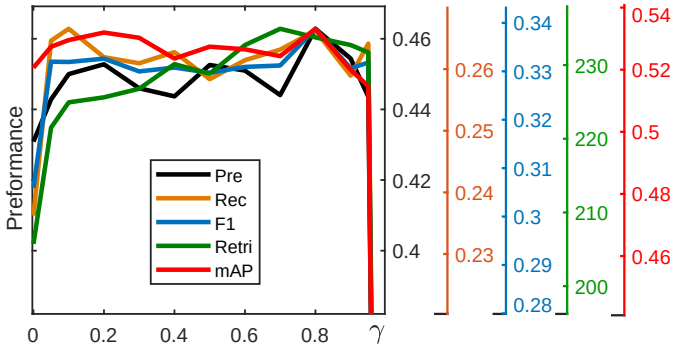

Fig. 6. Parameter sensitivity analysis: The performance of GCDN as γ changes from 0 to 1 in IAPRTC12 dataset. The result illustrates that evaluation metrics are high and stable when $\gamma \in [0.1, 0.9]$ which demonstrates the robustness and parameter insensitivity of our model.

performance but the improvement is not significant. To this end, we conclude that both GAN or CDN can effectively improve the performance independently, and the combination of the two components can let GAN and CDN help each other and dramatically improve and stabilize the performance.

To further demonstrate the effectiveness of the generative strategy, we add low-level Gaussian noise on original features. Table IV shows the classification performance based on various noise levels. It illustrates that the classifier obtains the highest performance if no noise is included in the features. To this end, we conclude that noise cannot increase the sample diversity and it could further destroy the feature structure and eliminate learning performance. This result indicates that the generator is indeed an effective approach to synthesize appropriate features to diversify and enlarge corresponding distributions in feature space.

In our model, $\gamma \in [0, 1]$ is a critical hyper-parameter, as introduced in Eq. (8), which balances the weights between $C_M(\cdot)$ and $C_{CDN}(\cdot)$. Now, we tune γ in $[0, 1]$ on IAPRTC12 dataset and Figure 6 shows the performance. We observe that our model achieves stable and highest performance when $\gamma \in [0.1, 0.9]$, which indicates the parameter insensitivity of our model. If γ is too close to 0, that means there is no control on $C_M(.)$ and $C_M(\cdot)$ could not be trained to output initial label prediction. By this way, the label relation matrix can only be considered as a regular feature extraction layer but without any reasonable logic which may decrease the generalization quality and cause overfitting issue. Thus, we can see the clear performance decreases when γ is close to 0. Meanwhile, if γ is too close to 1 would cause $C_{CDN}(\cdot)$ not be trained which significantly reduces the learning performance. These results demonstrate the necessity of jointly training C_{CDN} and C_M in our model. In the implementation, we empirically set $\lambda = 0.5$ to achieve the results of all datasets which denotes that 0.5 is appropriate enough for most applications without extra tuning procedure.

F. Discussion

We notice that our approach cannot achieve the best performance in AWA dataset in some metrics. We consider this in the following reasons. First, different from other datasets, AWA samples that belong to the same class share only one consistent semantic description (label vector), thus, it is difficult to comprehensively learn neither image-label mappings nor cross-label correlations; second, due to the consistent label issue, there are limited correlation information learnt by CDN to extend to other samples/classes. The result reveals the limitation of the proposed model but this scenario is unique which is not seen very often.

G. Image Annotation

We test the image annotation performance on SUN dataset and the result samples are shown in Figure 5. Figure 5 shows target images and the recovered labels are listed on the right. We set different colors to indicate different labels. The black font denotes correct labels. Considering there are more than 10 labels of an image in some cases, we only visualize and discuss the labels with the top 10 highest scores. The blue font indicates the missing annotations in ground truth but our model still promisingly recovers these labels based on our judgments. The red font denotes incorrect recovered labels from our

Fig. 7. Image retrieval result of SUN dataset in zero-shot scenario. Each row shows the images with the highest corresponding label score retrieved from the testing set. Green and red boxes indicate correct and incorrect retrieval, respectively. For each target label, we show the first incorrect retrieval result and its score ranking on the image right corner.

model. From the result, we can see that most recovered labels are correct with several discovered "new" labels. The results indicate that our model is effective and robust, which is able to recover the vast majority of labels from target images in high accuracy. Moreover, our model can find the missed and error labels of the ground truth.

H. Image Retrieval

We further test our approach in image retrieval scenario. Image retrieval is a visual search task that aims at retrieving target images from large-scale image sets. The target features can be visual, semantic or content descriptions. Image retrieval has a lot of real-world applications such as image search, person identification and data mining.

In the implementation, the trained classifier is used to predict labels of the testing images. Then we rank the images by each score of the label. When inputting a retrieval label, we can find the corresponding images from the ranking results. We run the test based on the same zero-shot settings, that means the retrieved classes don't exist in the training process. It is a more challenging task than conventional setting. Figure 7 shows the retrieved image samples. The left part lists the target label, and the right part shows the retrieved images. Green and red image edges indicate correct and incorrect retrieval results respectively. Since most top ranking images are correct, we intentionally select the first incorrect results of them and mark the images associated with the corresponding ranking numbers on bottom corner of each image. From 7, we can see that most images are correctly retrieved with only a few errors.

There are discussions for some phenomena we find in the results. First, the performance varies in different labels. For example, the retrieval performance of *metal* is better than *digging*. We observe that adjective and verb labels are more challenging than noun labels. Since it needs to analyze interactions between different features and more sophisticated context based structures are required for this challenge. Second, the model prefers specific scenes than others. Such as *sports*, the model prefers to retrieve all field scenes first instead of specific sport classes such as biking and swimming. Thus, more works can be done for these issues to get better retrieval performance.

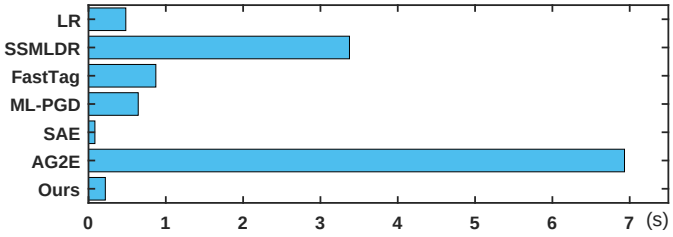

Fig. 8. Time consumption in inferring process which denotes the feasibility of our approach in large scale multi-label learning applications.

Figure 8 shows the time consumption of each method in the testing stage. Due to the simple feed-forward network structure and the usage of GPU acceleration, even if the computational cost is a little higher, our approach only spends an average of 0.12 seconds to infer 2081 testing samples which is the second fastest method. It indicates that our approach fits well for large-scale real-world applications.

V. Conclusion

We proposed a Generative Correlation Discovery Network (GCDN) for Multi-label Learning. Our model captures the visual distribution and generates diverse samples to fill up gaps between training and testing samples. A multi-label classifier is jointly trained based on both the generated and real samples to improve the robustness and accuracy. A simple but effective Correlation Discovery Network (CDN) is proposed to automatically explore the correlations across labels and dramatically improve the learning performance without any extra semantic information as prior knowledge. All networks are jointly trained in an end-to-end scenario. Our model is quantitatively and visually evaluated based on six datasets with four settings and significantly improves the performance. Ablation study demonstrates the necessities of all proposed strategies in our model for reaching high accuracy.

Acknowledgements: This research is supported by Samsung GRO program.

REFERENCES

[1] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.

[2] Y. Verma and C. Jawahar, "Image annotation by propagating labels from semantic neighbourhoods," *IJCV*, vol. 121, no. 1, pp. 126–148, 2017.

[3] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos, "Supervised learning of semantic classes for image annotation and retrieval," *TPAMI*, vol. 29, no. 3, pp. 394–410, 2007.

[4] T. Liu, D. Tao, M. Song, and S. J. Maybank, "Algorithm-dependent generalization bounds for multi-task learning," *TPAMI*, vol. 39, no. 2, pp. 227–241, 2016.

[5] L. Zhang, S. R. Wilson, and R. Mihalcea, "Multi-label transfer learning for semantic similarity," *arXiv preprint arXiv:1805.12501*, 2018.

[6] S.-J. Huang, W. Gao, and Z.-H. Zhou, "Fast multi-instance multi-label learning," *TPAMI*, 2018.

[7] Y. Song, L. Zhang, and C. L. Giles, "Automatic tag recommendation algorithms for social recommender systems," *ACM Transactions on the Web*, vol. 5, no. 1, p. 4, 2011.

[8] T. Liu and D. Tao, "Classification with noisy labels by importance reweighting," *TPAMI*, vol. 38, no. 3, pp. 447–461, 2015.

[9] B. Wu, W. Chen, P. Sun, W. Liu, B. Ghanem, and S. Lyu, "Tagging like humans: Diverse and distinct image annotation," in *CVPR*, 2018, pp. 7967–7975.

[10] B. Wu, F. Jia, W. Liu, B. Ghanem, and S. Lyu, "Multi-label learning with missing labels using mixed dependency graphs," *IJCV*, pp. 1–22, 2018.

[11] Z. Ding, M. Shao, S. Li, and Y. Fu, "Generic embedded semantic dictionary for robust multi-label classification," in *ICBK*, 2018, pp. 282–289.

[12] L. Von Ahn and L. Dabbish, "Labeling images with a computer game," in *SIGCHI*, 2004, pp. 319–326.

[13] M. Belkin, I. Matveeva, and P. Niyogi, "Regularization and semi-supervised learning on large graphs," in *Association for Computational Learning*, 2004, pp. 624–638.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NeurIPS*, 2014, pp. 2672–2680.

[15] F. Kang, R. Jin, and R. Sukthankar, "Correlated label propagation with application to multi-label learning," in *CVPR*, vol. 2, 2006, pp. 1719–1726.

[16] N. Ghamrawi and A. McCallum, "Collective multi-label classification," in *CIKM*, 2005, pp. 195–200.

[17] Y. Liu, R. Jin, and L. Yang, "Semi-supervised multi-label learning by constrained non-negative matrix factorization," in *AAAI*, vol. 6, 2006, pp. 421–426.

[18] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, T. Mei, and H.-J. Zhang, "Correlative multi-label video annotation," in *Multimedia*, 2007, pp. 17–26.

[19] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *ICML*, 2003, pp. 912–919.

[20] L. Wang, Z. Ding, and Y. Fu, "Low-rank transfer human motion segmentation," *TIP*, vol. 28, no. 2, pp. 1023–1034, 2019.

[21] M. Chen, A. Zheng, and K. Weinberger, "Fast image tagging," in *ICML*, 2013, pp. 1274–1282.

[22] W. Ge, S. Yang, and Y. Yu, "Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning," in *ICCV*, June 2018.

[23] X. Zhao, H. Li, X. Shen, X. Liang, and Y. Wu, "A modulation module for multi-task learning with applications in image retrieval," in *ECCV*, September 2018.

[24] P. Duygulu, K. Barnard, J. F. de Freitas, and D. A. Forsyth, "Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary," in *ECCV*, 2002, pp. 97–112.

[25] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," Tech. Rep. CNS-TR-2011-001, 2011.

[26] G. Patterson and J. Hays, "Sun attribute database: Discovering, annotating, and recognizing scene attributes," in *CVPR*, 2012, pp. 2751–2758.

[27] S. J. Pan and Q. Yang, "A survey on transfer learning," *TKDE*, vol. 22, no. 10, pp. 1345–1359, 2010.

[28] L. Wang, Z. Ding, and Y. Fu, "Learning transferable subspace for human motion segmentation," in *AAAI*, 2018, pp. 4195–4202.

[29] Q. Ma, H. Xia, G. Ma, Y. Xia, and C. Wang, "Improved stability and stabilization criteria for TS fuzzy systems with distributed time-delay," in *ICDMBD*, 2017, pp. 517–526.

[30] Q. Ma, L. Li, H. Xia, M. Yang, and G. Ma, "New results on stability and stabilization analyses for TS fuzzy systems with distributed time-delay under imperfect premise matching," in *ICICIP*, 2016, pp. 5–10.

[31] B. Wu, S. Lyu, and B. Ghanem, "Ml-mg: multi-label learning with missing labels using a mixed graph," in *ICCV*, 2015, pp. 4157–4165.

[32] C.-W. Lee, W. Fang, C.-K. Yeh, and Y.-C. Frank Wang, "Multi-label zero-shot learning with structured knowledge graphs," in *IEEE CVPR*, 2018, pp. 1576–1585.

[33] F. Tai and H.-T. Lin, "Multilabel classification with principal label space transformation," *Neural Computation*, vol. 24, no. 9, pp. 2508–2542, 2012.

[34] S.-F. Chen, Y.-C. Chen, C.-K. Yeh, and Y.-C. F. Wang, "Order-free rnn with visual attention for multi-label classification," in *AAAI*, 2018.

[35] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," *arXiv preprint arXiv:1812.04202*, 2018.

[36] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *NeurIPS*, 2016, pp. 3844–3852.

[37] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," *arXiv preprint arXiv:1611.04076*, 2016.

[38] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li, "Mode regularized generative adversarial networks," *arXiv preprint arXiv:1612.02136*, 2016.

[39] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *arXiv preprint arXiv:1703.10593*, 2017.

[40] L. Wang, Z. Ding, Z. Tao, Y. Liu, and Y. Fu, "Generative multi-view human action recognition," in *ICCV*, 2019.

[41] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[42] Z. Ding, M. Shao, and Y. Fu, "Generative zero-shot learning via low-rank embedded semantic dictionary," *TPAMI*, 2018.

[43] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *JMLR*, 2017, pp. 2642–2651.

[44] Z. Ding, Y. Guo, L. Zhang, and Y. Fu, "One-shot face recognition via generative learning," in *FG*, 2018, pp. 1–7.

[45] Z. Wang *et al.*, "Image quality assessment: from error visibility to structural similarity," *TIP*, 2004.

[46] G.-J. Qi, "Loss-sensitive generative adversarial networks on lipschitz densities," *arXiv preprint arXiv:1701.06264*, 2017.

[47] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *NeurIPS*, 2016, pp. 2234–2242.

[48] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv:1505.00853*, 2015.

[49] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015, pp. 448–456.

[50] M. Grubinger, P. Clough, H. Müller, and T. Deselaers, "The iapr tc-12 benchmark: A new evaluation resource for visual information systems," in *OntoImage*, 2006.

[51] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *TPAMI*, vol. 36, no. 3, pp. 453–465, 2014.

[52] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid, "Tagprop: Discriminative metric learning in nearest neighbor models for image annotation," in *ICCV*, 2009, pp. 309–316.

[53] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

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

[55] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[56] B. Guo, C. Hou, F. Nie, and D. Yi, "Semi-supervised multi-label dimensionality reduction," in *ICDM*, 2016, pp. 919–924.

[57] E. Kodirov, T. Xiang, and S. Gong, "Semantic autoencoder for zero-shot learning," in *CVPR*, 2017, pp. 3174–3183.

[58] L. Wang, Z. Ding, and Y. Fu, "Adaptive graph guided embedding for multi-label annotation." in *IJCAI*, 2018, pp. 2798–2804.

[59] L. Van Der Maaten, "Accelerating t-sne using tree-based algorithms," *JMLR*, vol. 15, no. 1, pp. 3221–3245, 2014.