Generative Multi-Label Correlation Learning

LICHEN WANG, Northeastern University, USA
ZHENGMING DING, Tulane University, USA
KASEY LEE, Northeastern University, USA
SEUNGJU HAN, Samsung Electronics, Korea
JAE-JOON HAN, Samsung Electronics, Korea
CHANGKYU CHOI, Samsung Electronics, Korea
YUN FU, Northeastern University, USA

In real-world applications, a single instance could have more than one labels. To solve this task, multi-label learning methods emerged in recent years. It is a more challenging problem for many reasons, such as complex label correlation, long-tail label distribution, and data shortage. In general, overcoming these challenges and bettering learning performance could be achieved by utilizing more training samples and including label correlations. However, these solutions are expensive and inflexible. Large-scale, well-labeled datasets are difficult to obtain, and building label correlation maps requires task-specific semantic information as prior knowledge. To address these limitations, we propose a general and compact Multi-Label Correlation Learning (MUCO) framework. MUCO explicitly and effectively learns the latent label correlations by updating a label correlation tensor, which provides high accurate and interpretable prediction results. In addition, a multi-label generative strategy is deployed to handle the long-tail label distribution challenge. It borrows the visual clues from limited samples and synthesizes more diverse samples. All networks in our model are optimized simultaneously. Extensive experiments illustrate the effectiveness and efficiency of MUCO. Ablation studies further prove the effectiveness of all the modules.

Additional Key Words and Phrases: Correlation learning, multi-label learning, image annotation, image retrieval.

ACM Reference Format:
Lichen Wang, Zhengming Ding, Kasey Lee, Seungju Han, Jae-Joon Han, Changkyu Choi, and Yun Fu. . Generative Multi-Label Correlation Learning. *J. ACM* 37, 4, Article 111 (August), 19 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Traditionally, single label scenario assumes one label per image instance. However, with the exponential growth of digital applications, real-world computer vision and machine learning tasks have found an increasing need to get multiple labels from an individual sample. For instance, a single image could contain multiple labels describing its category, size, color, shape, texture, and so

Authors' addresses: Lichen Wang, wanglichenxj@gmail.com, Northeastern University, Boston, Massachusetts, USA; Zhengming Ding, zding1@tulane.edu, Tulane University, New Orleans, Louisiana, USA; Kasey Lee, lee.kase@northeastern.edu, Northeastern University, Boston, Massachusetts, USA; Seungju Han, sj75.han@samsung.com, Samsung Electronics, Korea; Jae-Joon Han, jae-joon.han@samsung.com, Samsung Electronics, Korea; Changkyu Choi, changkyu_choi@samsung.com, Samsung Electronics, Korea; Yun Fu, yunfu@ece.neu.edu, Northeastern University, Boston, Massachusetts, USA.

0004-5411//8-ART111 $15.00
https://doi.org/10.1145/1122445.1122456

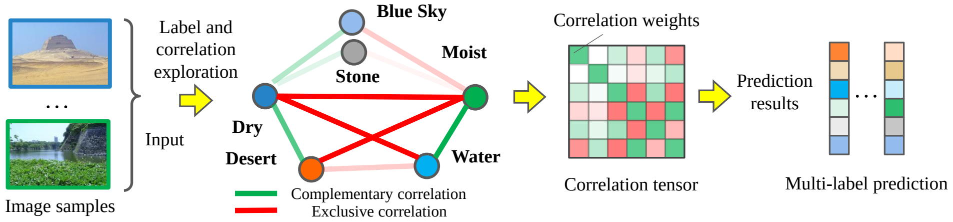

Fig. 1. Label correlation is the unique and crucial knowledge for predicting multi-labels effectively. For example, the label ***Dry*** and ***Moist*** cannot exist simultaneously, and they are exclusive to each other. Meanwhile, the label ***Water*** usually has a positive connection with ***Moist***. They are complementary to each other. Different labels could have various connections with other labels. We illustrate the complementary (green lines) and exclusive (red lines) correlations. Based on the correlation knowledge, the model could enhance the confidence of the multi-label predictions and eliminate incorrect predictions.

on, potentially reaching hundreds of labels. Due to this reason, multi-label learning methods are proposed to address the difficulties [2].

Limited training samples, long-tailed label distribution, and the latent connections between labels are the major challenges. The first challenge occurs commonly in most of the corresponding datasets. For instance, there are 18, 689 samples in the ESP dataset [40] and 268 candidate labels. However, there is only a small number of labels, 4.69, assigned to each image on average. Traditional multi-label learning approaches mainly deploy specifically designed constraints to reduce the negative influence of the overfitting issue. However, this strategy cannot be generalized. Others use auxiliary samples without labels to expand the label information in a semi-supervised setting [1]. However, this strategy can easily cause negative transfer when the auxiliary data differs significantly from the original dataset.

In addition, the challenge also comes from the long-tailed or uneven distribution of the labels. Specifically, certain labels (e.g., *"Man-made"*) occur commonly while others (e.g., *"Fire"*) rarely show up. This causes the training samples with "tail" labels to fail to cover the entire test space, and introduces potential biases for different labels that significantly decreases the number of positive predictions for "tail" labels. [52] investigates the negative impacts induced by imbalanced data on overall performance. [51] proposes a novel Distribution-Balanced strategy which re-balances the label co-occurrence weights. However, building large-scale, well-labeled, and task-specific datasets is much more expensive and difficult than building single-label datasets. Moreover, multi-label learning also has the unique characteristic of label correlation. As illustrated in Figure 1, the label *Dry* and *Moist* cannot exist simultaneously, and they are exclusive to each other. Meanwhile, the label *Water* usually has a positive connection with *Moist*. They are complementary to each other. Different labels could have various connections with other labels. Label correlation is the unique and crucial knowledge for predicting multi-labels effectively [48, 49]. However, obtaining the label correlations as the prior knowledge is difficult, since it requires expert knowledge, and it is not feasible to extend the correlations to other tasks, which limits the practicability of this strategy.

We proposed a Multi-label Correlation Learning (MUCO) method. Specifically, a multi-label generation strategy is designed to overcome the limited and long-tail distributed labels. The framework is shown in Figure 2. The generative model explores the visual distribution of the real images. Then, it borrows visual clues to generate more samples conditioned on the given labels. Meanwhile, a novel label correlation tensor is designed to effectively and explicitly extract the label correlations between the labels. The learned tensor is used to further fine-tune the prediction results. We listed our main contributions below:

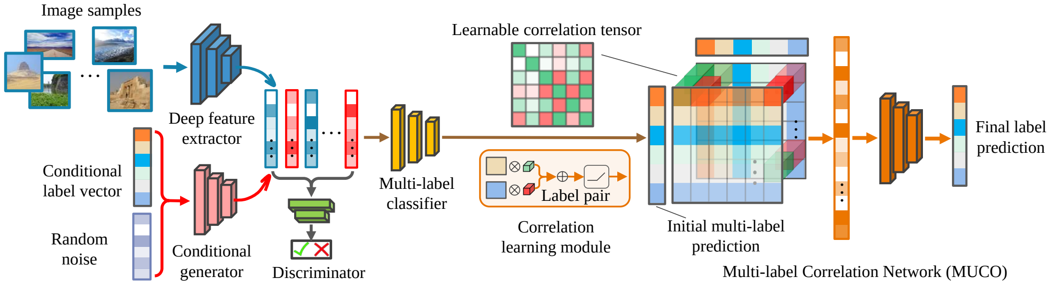

Fig. 2. Framework of our MUCO method. A multi-label generative strategy is deployed to explore the visual clues and diversify the training samples. It effectively mitigates the negative influence of the long-tail label distribution and limited training data challenges. A multi-label classifier is used to obtain initial prediction. A trainable label correlation tensor is proposed to learn the label correlations and further fine-tune the initial results. By this way, both the generated visual features as well as the label correlations could be fully explored. In the MUCO framework, all modules are simultaneously optimized, and the end-to-end training strategy makes our model practical for real-world applications.

- A multi-label generative network is specifically proposed to conditionally synthesize more diverse samples. It effectively addresses the long-tail label distribution difficulty as well as the training sample shortage issue.
- A specifically designed correlation learning module is proposed based on a correlation tensor which explicitly learns and utilizes the correlations.
- A multi-label prediction constraint and the feature label constraint is proposed to tune the diversity of the generation results. These constraints effectively stabilize the generative model and achieve more generalized performance.
- Experiments demonstrate the effectiveness of the model, and various ablation studies quantitatively and visually illustrate the contribution of each modules.

Our MUCO framework is a data driven approach which automatically explores the latent label correlations without any other semantic knowledge. Moreover, the MUCO deploys an end-to-end training strategy to optimized the all the weights jointly. To this end, it is easy and practical to deploy MUCO in any practical multi-label tasks. MUCO is an extension of the conference version of our work [44] with two major improvements. First, a novel label correlation exploration structure is proposed. It enhances the flexibility and capability to explore the correlations. Second, a tensor-based module is designed to learn the label correlations. Compared with other deep models, it provides interpretable correlation knowledge which could be helpful for further understanding and tuning the final prediction results. The experiments denote that MUCO gets higher evaluation results than other benchmark multi-label methods.

2 RELATED WORK

2.1 Multi-label Learning

The ultimate task of Multi-label learning is to explore the patterns between sets of labels for certain instances and is common in a lot of practical tasks, such as language models [9, 20, 25], image annotation [2, 15], and scene understanding [34]. Multi-label learning presents more challenges than conventional single-label classification tasks [2]. Multi-label learning has two types: supervised and semi-supervised [1, 2, 55]. Supervised learning include [5], which presents a way to reduce the negative influence of the noisy label, MEFF [8], which uses a fusion between multiple views to approach multi-label classification, and [53], which proposes a general modulation module that utilizes the related tasks to enhance the retrieval performance. These approaches all require relatively large-scale and well-labeled samples to work effectively. However, the scale of multi-label datasets [7, 32, 40, 41] are limited. Transfer learning [31, 35] is able to train models with fewer labeled instances while using large amounts of unlabeled, auxiliary data as supplement [1, 42, 55]. Nevertheless, effectiveness of these strategies depend on the compatibility of the source and target samples, which is hard to control. For example, if the data between the target and source domains is not similar, there will not be any useful information extracted from the auxiliary data. It can even hurt the performance due to the negative transfer. Long-tailed label distribution is another unique challenge of multi-label learning, which means certain labels occur commonly while others rarely show up. This causes the training samples with "tail" labels to fail to cover the entire test space, and introduces potential biases for different labels that significantly decreases the number of positive predictions for "tail" labels. [52] explores the negative influence of the data imbalanced issue, and proposed a RoBal framework which contains a scale-invariant classifier. [51] proposed a novel Distribution-Balanced strategy. It considers the label co-occurrence via a weights re-balancing strategy. [47] analyzes the potential impact of missing labels, noisy labels, and tail labels. A novel efficient low-complexity model is proposed. [3] proposed a X-Transformer framework, which fine-tunes transformer-based strategy. A LightXML framework is proposed in [14] which utilizes a sampling solution to handle the label imbalance issue. A ECLARE method is proposed in [28] which explores jointly considers the label correlations as well as the label text for predictions. A DECAF framework is proposed in [27] which fully explores the metadata of the label to get distinguished representations. Label correlation is also a crucial knowledge to improve performance. How to effectively explore the structural knowledge and utilize it to enhance the learning performance is always challenging task in machine learning fields [21, 22, 45]. [19, 23, 43, 49, 50] deploy graph-based methods by exploring the structural knowledge to introduce the label correlation. [48] uses a pre-existing label syntactic structure to discover correlation between different labels and deduce label noise. Label embedding [38] explores label relationships by projecting them into the semantic space. [6] uses attention and sequential (i.e., RNN and LSTM) modules to obtain the predictions. In these methods, the well-defined syntactical knowledge is crucial and necessary to obtain reasonable prediction results, and the knowledge is usually task-specific which is hard to be extended to other tasks. This issue makes the methods impractical.

Here, we propose a correlation learning approach in multi-label scenario. A generative model explores the visual distribution from the training samples and reuses these features across samples to generate more images for training. Our framework is more effective in obtaining the explainable label correlations and further improves learning performance.

2.2 Generative Adversarial Net (GAN)

Generative Adversarial Network [10] is the most popular and representative generation method. It is made up of two modules: a discriminator and a generator. The two modules are optimized in opposition; the generator aims to produce realistic samples to confuse the discriminator. On the other hand, the discriminator aims to differentiate between the generated and real samples. Both networks compete and improve until there are no distinguished differences between the samples for the discriminator. Different types of GANs are utilized for different applications. Least Squares GAN [24] bypasses the vanishing gradient challenge by using least squares in the training process. Mode Regularized GAN [4] greatly stabilizes the training of GAN models by proposing ways of regularizing the objective function. Cycle GAN [54] overcomes the absence of paired samples by proposing a structure that projects a visual sample across the target and source domains. Conditional GAN (CGAN) [26] expands on the traditional GAN model by adding further information as the condition, such as target categories. ACGAN [30], based on CGAN, promotes a classifier for generation which guides and stabilizes the generator's training process. However, ACGAN and CGAN were not designed for objective classification, as they use the perceptual test of human beings (e.g., MS-SSIM [46]) to demonstrate the diversity of the generated samples. Conditional Loss-Sensitive GAN (LS-GAN) [33] is used to assign single labels to target images and make the generated image more real with a loss function. Nevertheless, LSGAN is still hard and impractical for multi-label setting due to its optimization strategy.

Unlike previous works, our approach explores the generative model for multi-label classification. Our approach explores the latent connections across different visual components and increases the overall sample diversity for classification tasks rather than subjective human evaluation [46]. It is able to overcome challenges with limited multi-label training data and span the entire feature area.

3 OUR APPROACH

3.1 Motivation & Preliminaries

Formally, the features of the multi-label training data are given by $\{X_{tr}, Y_{tr}\}$. $X_{tr} \in \mathbb{R}^{d \times n_{tr}}$ is the feature matrix, where n_l is the number of training samples, d is the dimension of the feature vector, and each column $x_i \in \mathbb{R}^d$ represents one instance. Similarly, $Y_{tr} \in \mathbb{R}^{d_l \times n_{tr}}$ is the corresponding label matrix. d_l is the label dimension, and y_i represents the label vector corresponding to x_i. Our approach aims to recover the multi-label Y_{te} from the given testing samples X_{te}, by training samples $\{X_{tr}, Y_{tr}\}$ without any other prior knowledge. As we introduced in Section 1, the sample distribution in the visual space is considerably larger than the distribution in the label space. To this end, it is difficult to train a multi-label classifier directly without adding constraints. In addition, the label correlations between various samples could provide unique and informative clues. Exploring and utilizing these correlations is crucial for further improving the learning performance.

3.2 Our MUCO Framework

There are three major modules in our approach. A conditional generative module which includes a discriminator $D(\cdot)$, a generator $G(\cdot)$, and a multi-label classifier $C_M(\cdot)$. $C_M(\cdot)$ obtains initial prediction results from the given samples, and a multi-label correlation learning network $C_{MUCO}(\cdot)$ which explores the label correlation and finally determines the most accurate multi-label prediction results. In the generation stage, $G(\cdot)$ outputs the generated samples based on the input multi-label vector as its condition. The equation is shown below:

$$X_g = G(z|Y), \tag{1}$$

where z represents a vector of random noise and Y is a label vector extracted from real training instances. The discriminator, $D(\cdot)$, is then optimized to differentiate across the generated and real instances. The loss function is shown below:

$$L_D = E_{z \sim p_z(z)} \log(1 - D(G(z|Y))) + E_{X \sim p_X(X)} \log D(X|Y). \tag{2}$$

In this stage, the generator learns the visual distribution and components from the given samples. These components are used to generate more diverse samples.

Compared with the conditional generation task in the single-label scenario, multi-label samples contain more trivial details that contribute considerably to the final predictions. To this end, we deploy three constraints which seek more stable and fine-grained generation results. The first objective is the general adversarial training objective function with $D(\cdot)$. It lets $G(\cdot)$ generate visual samples to be as real as possible. The objective is shown below:

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

In addition, considering the sophisticated label combinations and the relatively small scale of the multi-label dataset, a classification similarity constraint is proposed. Inspired by the work of ACGAN [30], we pull the multi-label predictions of the generated samples to be as similar as the ground-truth samples. The loss function is illustrated below:

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

where $C_M(\cdot)$ is a initial multi-label classifier. In addition, due to the significant diversity of the input features, the generative model may not be stable enough for handling different scales of the training samples, especially the small-scale datasets. To this end, we further include a similarity constraint. It is a simple and straightforward objective which controls the similarities of the real and generated samples directly in the feature space, and the loss function is shown below:

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

The overall objective function of $G(\cdot)$ is combination of the above objectives:

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

where λ and α are set as the trade-off parameters.

The proposed generative model mainly addresses the feature diversity and shortage issues. Meanwhile, effectively exploring and utilizing the sophisticated label correlation knowledge is still a challenge. We introduce a correlation learning structure which specifically focuses on this challenge. First, a general multi-label classifier, $C_M(\cdot)$, is utilized to get the initial multi-label results. Specifically, in the training procedure, both the real and generated samples are used. The objective function is shown below:

$$L_{C_M} = \mu\|Y - C_M(G(z|Y))\|_{\mathrm{F}}^2 + \|Y - C_M(X)\|_{\mathrm{F}}^2, \tag{7}$$

where the two terms are the classification losses of the real and generated samples. Y is the real multi-label vector which is used to conditionally generate samples. μ is used to balance the contribution of generated and real data in the training process of $C_M(\cdot)$. $C_M(\cdot)$ is a simple and straightforward multi-layer neural network.

However, the structure of $C_M(\cdot)$ is too simple to extract the sophisticated label correlations, while we consider it still provides the initial prediction results. The continuous scores/confidences which contain extra information for further exploration. To this end, a specifically designed Multi-label Correlation Learning (MUCO) network is proposed. The network structure is illustrated in Figure 2. We assume the initially predicted label vector, f_i, is obtained by $C_M(\cdot)$:

$$f_i = C_M(x_i), \tag{8}$$

where $f_i \in \mathbb{R}^{d_l \times 1}$ is the prediction of a given instance x_i. Then, a horizontally padding is conducted on f_i to get a matrix F_i^H, where there are d_l of f_i be concatenated together:

$$F_i^H = \underbrace{[f_i, f_i, f_i, \cdots, f_i]}_{d_l}, \tag{9}$$

where $F_i^H \in \mathbb{R}^{d_l \times d_l}$. In addition, a similar vertical padding is further used on f_i to get matrix F_i^H:

$$F_i^V = \underbrace{[f_i, f_i, f_i, \cdots, f_i]}_{d_l}^\top = {F_i^H}^\top. \tag{10}$$

where $F_i^V \in \mathbb{R}^{d_l \times d_l}$. Given a position (i, j), $F_i^H(i, j)$ and $F_i^V(i, j)$ correspond to i-th and j-th labels testing scores.

To this end, a correlation tensor, $T_L \in \mathbb{R}^{d_l \times d_l \times 2}$, is further proposed to actively learn the weights between the label pairs. We can consider T_L is concatenated by two matrices $T_L = [T_L^H, T_L^V]$, where $T_L^H, T_L^V \in \mathbb{R}^{d_l \times d_l}$. Then, the correlation tensor and the label matrices are multiplied in element-wise:

$$F_i^C = \delta(F_i^H \circ T_L^H + F_i^V \circ T_L^V), \tag{11}$$

where $\circ$ is the multiplication operation in element-wise. $\delta(\cdot)$ is the ReLU or other activation functions, and $F_i^C \in \mathbb{R}^{d_l \times d_l}$ is the obtained matrix. By this way, each pair of initial predicted label scores are combined by the trainable tensor T_L, and the elements of T_L corresponding to the weights of the pairwise labels. By this way, T_L effectively and explicitly gets the correlation knowledge and the obtained F_i^C is a more comprehensive and correlation preserved representation. After that, we reshape F_i^C to a vector. Then, another fully-connected network, $C_{MUCO}(\cdot)$, is used to get the final results. The loss function is illustrated below:

$$L_{C_{MUCO}} = \sum_{i=1}^{n_l} \|y_i - C_{MUCO}(reshape(F_i^C))\|_2^2, \tag{12}$$

where $reshape(\cdot)$ is the reshape operation which reshapes a matrix to a vector. $y_i \in \mathbb{R}^{d_l \times 1}$ is the ground-truth vector. By this structure, the pairwise correlations could be effectively explored and used to obtain the more accurate final predictions.

In our model, both $C_M(\cdot)$ and $C_{MUCO}(\cdot)$ contribute to the final prediction results. To this end, we make $C_M(\cdot)$ and $C_{MUCO}(\cdot)$ be optimized simultaneously by adding their loss functions. The final objective is shown below:

$$L_C = \gamma L_{C_M} + (1 - \gamma) L_{C_{MUCO}}, \tag{13}$$

where $\gamma \in [0, 1]$ is the trade-off weight which balances the training of $C_M(\cdot)$ and $C_{MUCO}(\cdot)$. The joint training strategy allows both networks to be flexible and compatible with each other and achieve the highest performance. In summary, $C_M(.)$ provides the rough prediction results by exploring the feature-label relations, and $C_{MUCO}(.)$ further explores the label-label correlations to refine the initial results.

In our implementation, the discriminator $D(\cdot)$ is a fully-connected network with three layers. The ReLU [29] is deployed as the activation function in the first layer, and a mini-batch [36] operation is set as the second layer, the final layer is with the Sigmoid activation function. Moreover, $C_M(\cdot)$ and $C_{MUCO}(\cdot)$ are 2-layer fully-connected networks. In its first layer, ReLU activation is deployed for non-linear projection, and the Sigmoid activation before output is used in the final layer for prediction output. For hyper-parameters, we empirically set $\mu = 1$, which expects the numbers of the real and generated samples to be evenly used in the training procedure. In addition, we set $\gamma = 0.5$, which indicates that the $C_M(\cdot)$ and $C_{MUCO}(\cdot)$ have the same weights. The detailed parameter sensitivity analysis and discussion are provided in Section 4.

3.3 Discussion

Our MUCO approach differs from the conventional generative model in multiple ways. First, MUCO utilizes multi-label knowledge as the conditions to synthesize additional samples. While, other models are traditionally used in the easier single-label scenario. Second, our approach can be better applied in various real life applications, as the label correlations are learned in the training stage without the need for additional knowledge (e.g., word/semantic embedding, hierarchical correlation). Third, the model is efficient in the testing stage compared with other graph-based or subspace learning approaches. In addition, the we trained our model in the end-to-end protocol, which makes the model compatible with diverse tasks. The end-to-end strategy requires more stabilization designs including the similarity constraint and the classification similarity loss. In addition, a more refined learning rate tuning process of the generator and discriminator may be needed in the model training process. The MUCO module could also be trained in a two stages pipeline where the first stage is training the generative networks and obtaining the stable generated samples. After that, the second stage is for the MUCO training where all networks are jointly trained. This strategy could make the training procedure and the learning performance more stable.

We further provide the spatial complexity of our model which is $O(d_l^2 + d_l^3)$. Specifically, the spatial complexity of the correlation tensor is quadratic to the number of label candidates, which is $O(d_l^2)$. In addition, the following $C_{MUCO}(\cdot)$ network where the input is the reshaped F_i^C, and the output is the final prediction results, which derives the $O(d_l^3)$ complexity. Thus, the overall complexity of the MUCO module is $O(d_l^2 + d_l^3)$. Our current model is able to handle the scenario with more than 300-dimension labels, which could be deployed in various applications. However, if there are a large number of labels, it would lead to potential high spatial complexity especially in extreme multi-label scenario when millions of labels exist. We proposed two potential solutions for this issue. First, the sparse constraint is a solution, since most of the pairwise labels are not correlated, which means there are only a small partial of labels that are correlated with another specific label. Second, the labels could be separately considered based on the semantic connections via the assistant of general natural language processing approaches, where the potential correlated labels could be explored locally, and then merged with other predictions for the final prediction.

4 EXPERIMENTS

Here, we utilize six multi-label datasets to test MUCO. State-of-the-art benchmarks are also evaluated. Four different settings are deployed including traditional multi-label classification, zero-shot multi-label classification, image retrieval, and image annotation. Below is the brief introductions of the datasets, and the dataset summary is listed in Table 1.

- **Corel5K Dataset** [7] includes 4, 500 training samples and 499 testing samples. The samples are extracted from the Corel CD Photo dataset. There are 260 candidate labels and the average of 3.40 labels from each sample.
- **ESP Game Dataset** [40] deployed a human-machine interactive system for labeling. The labeling process was designed to be similar to a computer game. ESP includes 18, 689 training instances and 2, 081 instances for testing. There are 268 candidate labels, and 4.69 labels were set to the instance on average.
- **IAPRTC-12 Dataset** [11] is used for the image retrieval task in the cross-language scenario. This dataset contains 19, 627 samples including animals, actions, landscapes, and other objects. These samples consist of 17, 665 instances for training and 1, 962 instances in the testing set. There are 291 candidate labels, with an average of 5.72 labels per sample.
- **SUN Dataset** [32] contains scene images captured in various locations, such as playground, classroom, and street. It has 717 different scene classes associated with 102 candidate labels, with an average of 6.31 labels per sample. For the conventional setting, we randomly selected 6, 387 samples for training and 6, 513 samples for testing. In the zero-shot scenario, it contains 12, 900 training images and 1, 440 testing images as default.
- **CUB Dataset** [41] is a bird database containing 200 bird categories. Each image has an average of 31.39 labels. In the conventional setting, more than 4, 000 images are set as training and testing by random selection. For zero-shot setting, the numbers of training and testing samples are 8, 842 and 2, 946.
- **AWA Dataset** [18] has around 30, 000 animal images of 50 different species. Each instance has an average of 15 labels, and there are 85 candidate labels. The values in the label vector are continuous in $[0, 100]$. Similarly, there are 12, 154 and 12, 141 images for training and

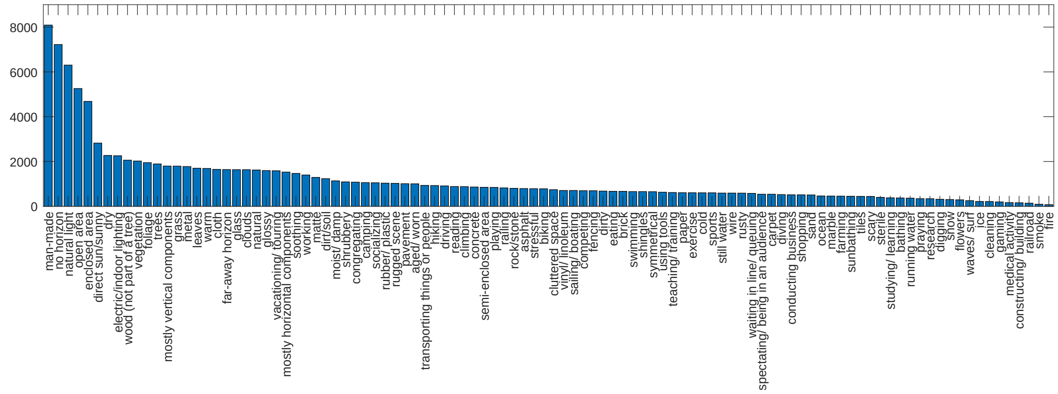

Fig. 3. Long-tail label distribution of SUN dataset

Table 1. Datasets statistical summary

Datasets	Setting	Training	Testing	Label-dim	Ave-label
Corel5K [7]	Conventional	4,500	499	260	3.40
ESP Game [40]	Conventional	18,689	2,081	268	4.69
IAPRTC-12 [11]	Conventional	17,665	1,962	291	5.72
SUN [32]	Conventional	6,387	6,513	102	6.31
	Zero-shot	12,900	1,440		
CUB [41]	Conventional	4,374	4,468	312	31.39
	Zero-shot	8,842	2,946		
AWA [18]	Conventional	12,154	12,141	85	15.00
	Zero-shot	24,295	6,180		

testing respectively in the general setting, and 24, 295 and 6, 180 images for training and testing in the zero-shot scenario.

In multi-label scenario, the long-tail label distribution is the unique challenge. We plot the histogram of the label distributions of SUN dataset in Figure 3. It explicitly illustrates the significant label imbalance situation. Specifically, the most common labels, *man-made*, exist in more than 8000 samples. However, there are only 74 samples which contain the label of *fire*.

4.1 Experimental Setup

For fair comparison, we follow the same feature extraction protocol proposed in [12] for ESP Game, IAPRTC, and Corel5K which contains 15 different visual descriptors. For the rest fo the datasets, we use the pre-trained VGG network [37] to obtain the deep representations. Since the range of AWA dataset label is $[0, 100]$, thus we multiply 100 after the Sigmoid activation output. As displayed in Figure 2, the random noise vector and the conditional multi-label vector are concatenated together as the input to the generator. Then, for the rest of our model, λ, which is used to limit the feature scales, is set to 20 for handcrafted representations [12] and 5 for VGG [37] deep representations. In addition, α is set to 0.01 empirically. Optimizer is set as the ADAM [16] method. Consider the different convergent speeds of different modules, for $D(\cdot)$, $C_{MUCO}(\cdot)$, and $C_M(\cdot)$ networks, we use 0.001, 0.00002, and 0.00002 respectively.

In the training procedure, all networks are assigned with random weights without any pre-training step. We observed a slight performance vibration in the beginning as the iteration increased. We consider this is due to the generator was not well trained. The vibration disappears soon and the performance becomes stable after around 5000 iterations. This end-to-end strategy requires more stabilization designs including the similarity constraint and the classification similarity loss. As we discussed above, a two-phases training strategy where the generative module is firstly trained followed by the rest of the networks is also a potential solution. We tested the methods 5 times on the training and testing datasets, which are randomly separated from the samples to be relatively even in sample size, then the average performance is reported. In our implementation, the TensorFlow associated with the GPU acceleration are deployed.

4.2 Multi-label Classification

Seven state-of-the-art multi-label classification methods are set as baselines, and the details are introduced below:

- **Least Square Regression (LR)** is a linear regression approach which aligns the labels with the corresponding features. We set it as a basic baseline for the evaluation.
- **Semi-Supervised Multi-Label Dimension Reduction (SSMLDR)** [13] is a semi-supervised approach. It improves the robustness and accuracy of the model with a label propagation strategy that uses both the labeled and unlabeled samples.
- **Fast Image Tagging (FastTag)** [5] is an efficient approach. It aims to infer complete labels from a list of incomplete ground-truth labels. Two linear projections are used to detect the missing label and obtain final prediction results respectively.
- **Multi-Label Mixed Graph (ML-PGD)** [50] infers label set with missing labels, but through a label dependency model constructed through class correlations.
- **Semantic AutoEncoder (SAE)** [17] recovers labels using a specifically designed auto-encoder. The feature is projected to label space, and the labels are projected to the original space. The weights are shared between features and labels.
- **Adaptive Graph Guided Embedding (AG^{2}E)** [42] recovers multiple labels in the semi-supervised scenario. A graph is proposed to adaptively explore and transfer the similarities knowledge between label space and feature space.
- **Generative Correlation Discovery Net (GCDN)** [44] explored a discovery network to reveal the relations of labels. Generation strategy is utilized to address the overfitting issues.

Since SSMLDR and AG^{2}E are semi-supervised learning methods, the testing samples are set as the unlabeled samples to test the results. For consistent comparison, we use the same metrics adopted in [12]. We calculated the precision metric (Pre) P and the recall metric (Rec) R, where $P = \frac{t_p}{t_p+f_p}$ and $R = \frac{t_p}{t_p+f_n}$. t_p is truth-positive. f_n and f_p denotes the false negative and the false positive predictions. For easier comparison, we calculated the harmonic mean of the precision, F1-score (F1), where $F_1 = 2\frac{P \times R}{P+R}$. We provide the mAP (mean average precision) which used in [50]. The non-zero predicted labels (non-zero recall, N-R) is also reported in our experiments.

Table 2 illustrates the performance in conventional setting. As shown by the results, our MUCO framework achieves higher performance compared with other methods, it obtains up to 5.2% and 7.8% improvements in precision and recall. The results demonstrate its accuracy and robustness. Additionally, we further tested our model following the work of [50], which proposes an augmented and more complete ground-truth labels for the ESP and Corel5K. [50] completes the labels of ESP datasets from 4.69 labels per sample to 7.27 per sample, and Corel5K dataset from 3.40 per sample to 4.84 per sample. The results, shown in Table 3, shows that the MUCO method still obtains higher accuracy then most benchmarks.

Table 2. Performance of conventional setting

Data	Method	Pre	Rec	F1	N-R	mAP	Data	Method	Pre	Rec	F1	N-R	mAP
Corel	Regression	0.2859	0.3211	0.3025	128	0.3630	SUN	Regression	0.6209	0.1473	0.2457	102	0.6807
	SSMLDR	0.2741	0.3366	0.3022	143	0.3410		SSMLDR	0.6879	0.1700	0.2726	102	0.6723
	FastTag	0.3123	0.3657	0.3369	143	0.3871		FastTag	0.6816	0.1473	0.2457	102	0.6914
	ML-PGD	0.2575	0.2911	0.2732	122	0.3727		ML-PGD	0.7110	0.1614	0.2631	101	0.7087
	SAE	0.2962	0.3442	0.3184	141	0.3823		SAE	0.7183	0.1638	0.2668	98	0.7012
	AG^{2}E	0.3011	0.3520	0.3245	**157**	0.3568		AG^{2}E	0.7685	0.1765	0.2871	99	0.6778
	GCDN	**0.3335**	0.3714	0.3514	148	0.4417		GCDN	0.7985	0.1835	0.2985	102	0.7093
	Ours	0.3230	**0.3913**	**0.3539**	151	**0.4523**		Ours	**0.8013**	**0.1903**	**0.3076**	**102**	**0.7126**
ESP	Regression	0.3793	0.2038	0.2653	215	0.3440	CUB	Regression	0.2010	0.0239	0.0428	157	0.0638
	SSMLDR	0.3298	0.1885	0.2399	226	0.3156		SSMLDR	0.3410	0.0473	0.0832	178	0.2329
	FastTag	0.4011	0.1927	0.2617	208	0.3904		FastTag	0.2147	0.0359	0.0615	167	0.3144
	ML-PGD	0.3239	0.2012	0.2482	210	0.4077		ML-PGD	0.3334	0.0451	0.0794	155	0.3288
	SAE	0.3861	0.1743	0.2402	194	0.3842		SAE	0.3383	0.0514	0.0908	196	0.3255
	AG^{2}E	0.3548	0.1525	0.2133	213	0.3730		AG^{2}E	0.3409	0.0531	0.0911	190	0.3106
	GCDN	0.4032	0.2178	0.2828	239	0.4327		GCDN	0.3718	0.0541	0.0944	214	0.3561
	Ours	**0.4224**	**0.2288**	**0.2969**	**239**	**0.4410**		Ours	**0.3912**	**0.0583**	**0.1014**	**223**	**0.3762**
IAP	Regression	0.4287	0.2041	0.2765	199	0.4211	AWA	Regression	0.8798	0.0821	0.1500	75	0.8626
	SSMLDR	0.3491	0.2520	0.2927	229	0.3981		SSMLDR	0.7812	0.0858	0.1546	67	0.8346
	FastTag	0.4346	0.2267	0.2980	227	0.4596		FastTag	0.7861	0.0949	0.1694	72	0.8791
	ML-PGD	0.4132	0.2441	0.3011	230	0.4674		ML-PGD	0.5395	0.0635	0.1136	57	0.9121
	SAE	0.3537	0.2282	0.2774	213	0.4309		SAE	0.9683	**0.0957**	**0.1742**	73	**0.9397**
	AG^{2}E	0.3829	0.2330	0.2897	229	0.4353		AG^{2}E	0.8483	0.0827	0.1507	73	0.9033
	GCDN	0.4732	0.2648	0.3396	237	0.5295		GCDN	0.9716	0.0871	0.1599	83	0.9291
	Ours	**0.4812**	**0.2653**	**0.3420**	**237**	**0.5315**		Ours	**0.9787**	0.0894	0.1638	**83**	0.9341

Table 3. Performance on augmented label sets

Data	Methods	Pre	Rec	F1	N-R	mAP	Data	Methods	Pre	Rec	F1	N-R	mAP
Corel-A	Regression	0.2842	0.2304	0.2545	103	0.3762	ESP-A	Regression	0.3848	0.1256	0.1894	178	0.3913
	SSMLDR	0.3036	0.2791	0.2908	134	0.3660		SSMLDR	0.3253	0.1697	0.2231	202	0.3357
	FastTag	0.3329	0.3145	0.3234	136	0.4127		FastTag	0.3886	0.1531	0.2197	196	0.4254
	ML-PGD	0.3245	0.3011	0.3124	140	0.4275		ML-PGD	0.3713	0.1184	0.1795	162	0.4211
	SAE	0.3168	0.3037	0.3101	128	0.4192		SAE	0.3153	0.1425	0.1966	156	0.4050
	AG^{2}E	0.3273	0.3172	0.3221	143	0.3985		AG^{2}E	0.3518	0.1492	0.2095	196	0.4030
	GCDN	0.3438	**0.3219**	0.3325	138	0.4773		GCDN	0.4772	0.1944	0.2763	225	0.4436
	Ours	**0.3612**	0.3081	**0.3325**	**144**	**0.4792**		Ours	**0.4827**	**0.1953**	**0.2781**	**225**	**0.4512**

4.3 Zero-shot Multi-label Classification

MUCO is tested in the zero-shot classification scenario, where categories of testing are not in the training. This scenario is more difficult due to the significant distribution gaps between the training and setting samples. We evaluate the results on the AWA, SUN, and CUB databases. These datasets are split into training and testing samples by default. The SUN database has 645 training classes and 72 testing classes. The AWA database contains 40 training categories and 10 testing categories. Finally, The CUB database contains 150 training classes and 50 testing classes. Table 4 illustrates the performances in this setting. It denotes that our method outperforms other benchmarks. This zero-shot setting further demonstrates the robustness of our model when even unseen categories are set as inputs. We consider this advantage makes our model more practical for deploying in different applications without further task-specific modifications.

4.4 Model Analysis

The generative strategy is crucial in our approach. To further analyze its effectiveness, we visualize 10 unseen predicted labels of CUB classes with the visual features as the input for generation. In

Table 4. Performance of zero-shot multi-label classification

Data	Method	Pre	Rec	F1	N-R	mAP
SUN	Regression	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
	AG^{2}E	0.7125	0.1618	0.2637	88	0.6693
	GCDN	0.7531	**0.1857**	0.2979	101	0.6911
	Ours	**0.7825**	0.1851	**0.2994**	**101**	**0.7032**
CUB	Regression	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
	AG^{2}E	0.2808	0.0481	0.0821	163	0.2693
	GCDN	0.3091	0.0488	0.0843	179	0.3264
	Ours	**0.3319**	**0.0507**	**0.0879**	**190**	**0.3345**
AWA	Regression	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**
	AG^{2}E	0.8247	0.0811	0.1476	71	0.8874
	GCDN	0.9249	0.0804	0.1480	83	0.8784
	Ours	**0.9382**	0.0813	0.1542	**83**	0.8815

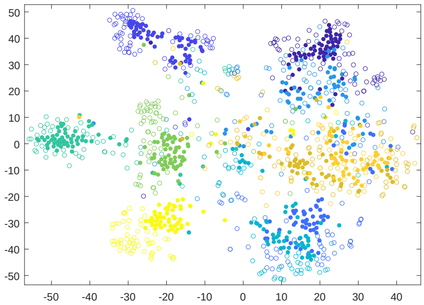

Fig. 4. The visualization of the real and generated samples. Solid circle denotes real samples and hollow circle denotes generated samples, and the different colors indicate different categories. The generated samples effectively diversity the visual distribution around the real samples which illustrate the effectiveness of the generative strategy.

our experiments, t-SNE [39] operation is deployed to project the high dimensional samples to the 2-dimensional for visualization. The results are shown in Figure 4, where the solid circles denote the real samples, the hollow circles denote the generated samples, and different colors represent different categories. From Figure 4, we can conclude that both the generated samples are close to the corresponding read samples, which effectively extend the feature distribution.

A few ablation studies are done to evaluate the improvements of different combinations of the proposed modules. Specifically, we remove the modules including the generative module, the multi-label correlation learning module, and both. In addition, we also tested our model with a basic network structure. In the experiment, a vanilla fully-connected network is used to replace MUCO module, where the input is the initial predicted label vector, f_i, and the output is the final

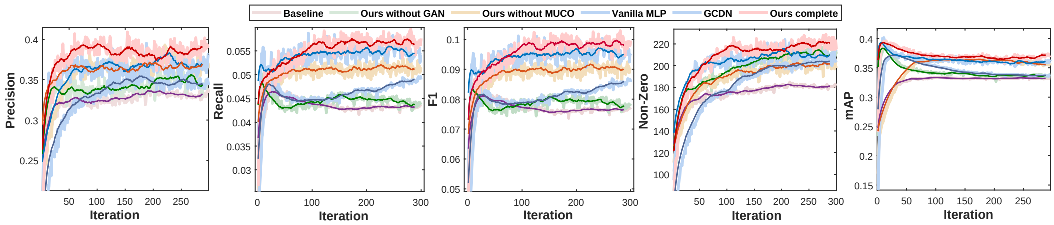

Fig. 5. Ablation study of our approach. The performances in different iterations are shown as curves. Different colors denote different models. The thick light color curve is the exact performance of each iteration, and the darker color curve is the smoothed result which provides a more clear performance comparison. Specifically, the **red** curve is our complete MUCO framework, the green curve ablated the generative module, the **yellow** curve ablated the MUCO network, and the **brown** curve is the baseline which removed both the generative and MUCO modules. In addition, we also show our previous GCDN approach (**blue** curve), and the approach that replaces the MUCO module with a basic fully-connected network (**purple** curve). We can observe that our complete MUCO framework clearly and considerably outperforms others.

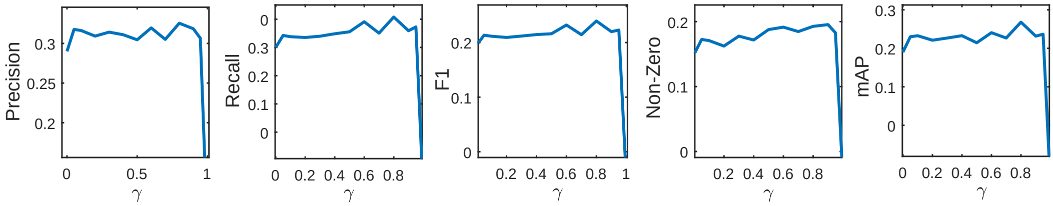

Fig. 6. The performance with different values of γ, where γ is proposed to tune the weights of $C_M(\cdot)$ and $C_{MUCO}(\cdot)$. The results is obtain in the zero-shot scenario on the CUB database. This shows that the performance remains stable until $\gamma > 0.95$, which shows the robustness of our MUCO framework.

prediction results. The performance of our previous work [44] is also visualized. The experimental results on the CUB dataset are illustrated in Figure 5. It shows the performances of 5 metrics in different iterations. The colors indicate different settings.

We can see that the complete MUCO module is robust and obtains the best compared with other ablation models. This fact concludes that all the networks in our model are effective. Specifically, the performance decreases if the generative strategy is removed. We assume it is due to the considerable imbalanced and long-tail labels and that the generative model is considerably helpful in this scenario. The simple network achieves good performance, but is still not comparable with our MUCO module. It denotes the effectiveness of the MUCO structure in capturing the label correlation knowledge and improving the learning performance. In addition, there is an interesting phenomenon where the mAP performance decreases slightly as the iteration increases. We assume it is due to the network being not well-trained.

As illustrated in Figure 5, the generative module is crucial for improving the learning performance. In our above experiment, we set $\mu = 1$, which means the weight of the generation is the same as the real data for training the classifier. In this ablation experiment, we analyze the parameter sensitivity of μ. We tune μ in different value and the results are visualized in Figure 7 and listed in Table 6. We observe that as the weight increases, the performance becomes better, and the performance tends to be stable as the weight is around 1. When μ is too large, the performance decreases a little. The result indicates that the generated samples are effective but still not as helpful as real data. From

Table 5. MUCO module ablation study

Ablation networks	Pre	Rec	F1	N-R	mAP
None	0.3194	0.0435	0.0753	174	0.3216
1-layer MLP	0.1674	0.0394	0.0638	174	0.2187
2-layer MLP	0.3180	0.0470	0.0818	177	0.3233
3-layer MLP	0.3117	0.0480	0.0833	178	0.3225
Ours	**0.3319**	**0.0507**	**0.0879**	**190**	**0.3345**

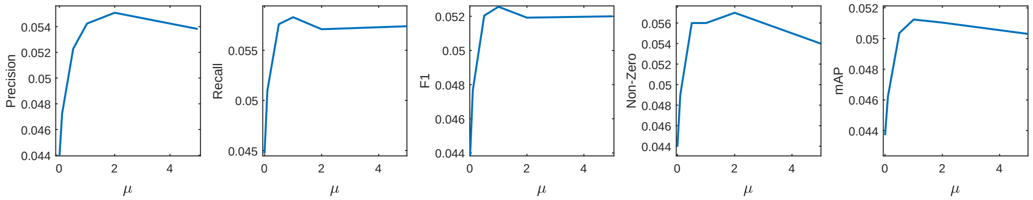

Fig. 7. μ is used to tune the objective weights between the real samples and the generated samples. We observe that when μ is around 1 (which means the even weights of the generated and real samples), MUCO achieves the best performance.

Table 6. Multi-label learning performance with different weights on the generated samples

Generative Weights μ	Pre	Rec	F-1	N-R	mAP
0.0	0.3388	0.0447	0.0790	206	0.3385
0.1	0.3562	0.0510	0.0892	211	0.3511
0.5	0.3814	0.0576	0.1001	218	0.3718
1.0	0.3912	**0.0583**	**0.1014**	218	**0.3762**
2.0	**0.3954**	0.0571	0.0998	**219**	0.3752
5.0	0.3891	0.0574	0.1000	216	0.3715

the results we conclude that μ being set around 1 is a reasonable and appropriate hyper-parameter for most of the datasets.

$\gamma \in [0,1]$ is another important hyper-parameter that balances the weights between $C_M(\cdot)$ and $C_{MUCO}(\cdot)$. To test the parameter sensitivity, we change γ from $[0,1]$ on the CUB dataset. Figure 6 displays our performance results. By observation, our model is stable and high-performing throughout $0 < \gamma < 0.95$, indicating the general parameter insensitivity of our model. In the extreme cases, if γ is around 1, $C_{MUCO}(\cdot)$ would not be trained, which leads the considerably performance decrease. Following a similar line of reasoning, if γ is around 0, $C_M(.)$ is not given any control, resulting in an inability for it to be optimized based on the supervision guidance. If $C_M(.)$ is not specifically trained, the label correlation tensor is degraded to a regular feature extraction layer, which would decrease to a general multi-layer network. With this logic, it is understandable why the performance decreases when γ is close to 0. These results illustrate the necessity for training both $C_M(\cdot)$ and $C_{MUCO}(\cdot)$ jointly with multi-label supervision in our model. In the experiments, results are achieved for all datasets with an empirically set value of $\lambda = 0.5$, implying that 0.5 is robust to various datasets and no extra parameter tuning is required.

The essential insight of MUCO is to extract and utilize the correlations of labels via the label correlation tensor. Technically a conventional multi-layer network could also achieve this goal, so we replaced the MUCO module with a fully-connected neural network. Similarly, the initial

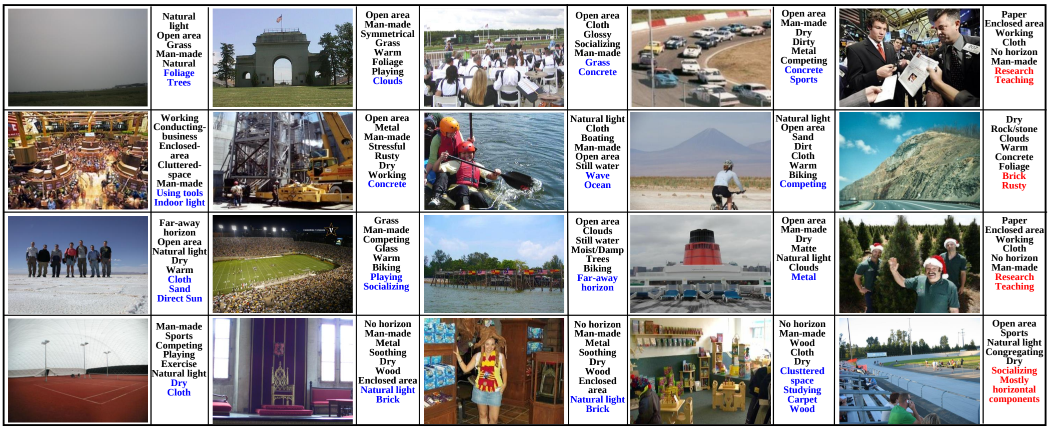

Fig. 8. Image annotation case studies. For each sample, the target image and the predicted multiple labels are illustrated. We utilize the **Red** font to show the incorrect results and the **Black** font to show the right label predictions. In addition, some labels are missing in ground-truth while they are still reasonable for the corresponding samples. We marked them as missing labels by **Blue** font. From the result we conclude that our MUCO framework is able to robustly predict the given samples even in the more difficult zero-shot scenario.

prediction label vector is set as input and the outputs are the final predictions. The results are listed in Table 5, where we tested networks with 1, 2, and 3-layers. From Table 5, we can see that the performance of 1-layer network is low, and the 2-layer and 3-layer networks achieve higher and similar performance. We conjecture that a one layer network is not capable of learning the complicated correlations, while the performances of 2 and 3 layer networks are saturated. The results demonstrate that our MUCO framework structure is more flexible and effective in capturing correlation information than naive multi-layer fully-connected networks.

The MUCO module shares some similarities in high-level compared with the attention module. The attention-based methods explores either the local or global connections in visual space, which find the correlations between different labels and fine-tunes the overall performance. However, our label correlation learning strategy explicitly creates the pairwise connections. It is a more specific-designed module which focuses and only focuses on the label correlation challenges, which would not be influenced by other aspects such as noise in visual space. To this end, we consider our approach to be a more effective module for multi-label learning.

4.5 Image Annotation

Figure 8 shows image annotation results of the SUN dataset. Target images are displayed with their corresponding recovered labels on the right. Label colors correspond to different levels of correctness. Blue font indicates labels that are correct by our judgements, but not included in the ground truth. Black font denotes correct labels according to ground truth, and red font indicates incorrectly recovered labels. We can observe that most of the labels are correctly recovered. Moreover, there are some new "correct" labels that emerged in the prediction results. These labels are initially indistinguishable in the visual space, but the learned correlation knowledge enhances their scores and allows them to eventually be discovered. The prediction results denote the robustness and effectiveness of the MUCO framework.

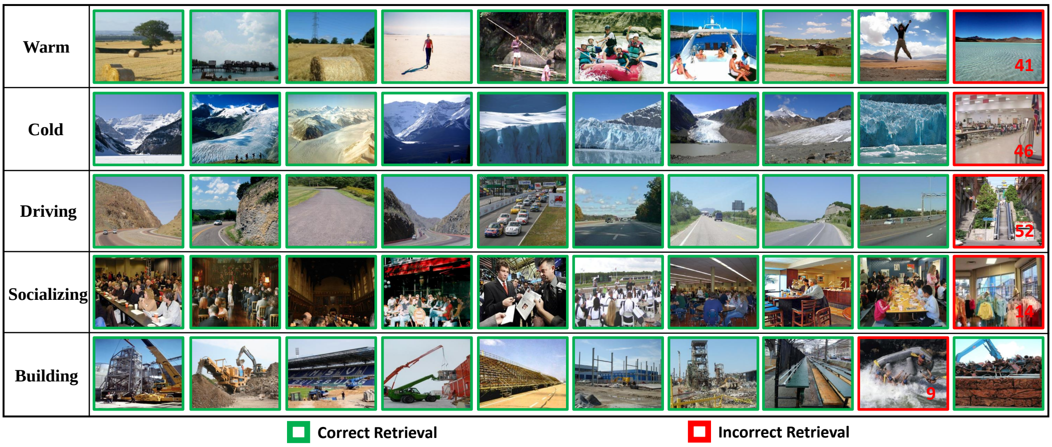

Fig. 9. Image retrieval case study. The left is the expected labels, and the rest are the retrieved images. To intuitively illustrate the effectiveness of our approach, the categories of the images are not shown in the training stage. The red number denotes the first incorrect retrieval of the given label. The results explicitly show the MUCO effectiveness and its potential for practical and large-scale applications.

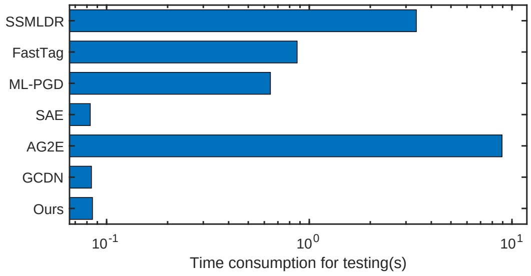

Fig. 10. Comparison of the running time in the testing stage. Due to the parallel computing via GPU acceleration, MUCO achieves similar efficiency compared with the fastest methods. It illustrates the the feasibility of our MUCO framework in large scale tasks.

4.6 Image Retrieval

We analyze the retrieval results of MUCO. This task retrieves target samples from a given set. In general, the samples used in retrieval can be either visualizations, signals, or sentences. Image retrieval is widely deployed in application, such as searching engine, data mining, and human identification. For our experiment, we are performing retrieval based on labels describing the target images. The MUCO model first predicts labels to the target images. Afterwards, we can use the ranking results to find the corresponding images for an inputted retrieval label. We test with zero-shot settings. Figure 9 listed the extracted images. The target labels are listed on the left with its retrieved images on the right. Because most of the top ranking samples were retrieved correctly, we have intentionally selected the first incorrect results for each label to be displayed in the figure. Each incorrect image is labeled with its ranking number in the bottom right corner. From Figure 9, we can see that most images are retrieved correctly.

4.7 Time consumption

Since our model is a deep learning model which could be computed in parallel via GPU acceleration, the speed of our approach is fast compared to other conventional methods. To quantitatively analyze the speed, we tested all the methods in the ESP dataset to infer 2081 testing samples. The time consumption of all the benchmarks are shown in Figure 10. Our MUCO framework achieves similar speed compared with the fastest methods (i.e., SAE). To this end, our MUCO framework is a practical method for large-scale multi-label applications.

5 CONCLUSION

We designed a novel Generative Multi-label Correlation Learning (MUCO) network. MUCO deploys the generative strategy, which borrows the visual components and generates more diverse samples for the training procedure. It effectively addresses the small-scale dataset limitation and the long-tail label distribution challenges. In addition, a specifically designed correlation learning structure associated with the trainable correlation tensor is used to explore the connections across pairwise labels, which effectively refines and considerably improves the prediction performance. In our model, all the weights are optimized simultaneously in an end-to-end setting to obtain the best performances and practicability. Multiple ablation analyses illustrate the contribution and efficiency of the proposed modules.

REFERENCES

[1] Mikhail Belkin, Irina Matveeva, and Partha Niyogi. 2004. Regularization and semi-supervised learning on large graphs. In *proceedings of Association for Computational Learning*. 624–638.

[2] Matthew R Boutell, Jiebo Luo, Xipeng Shen, and Christopher M Brown. 2004. Learning multi-label scene classification. *Pattern Recognition* 37, 9 (2004), 1757–1771.

[3] Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit S Dhillon. 2020. Taming pretrained transformers for extreme multi-label text classification. In *proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3163–3171.

[4] Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. 2016. Mode regularized generative adversarial networks. *arXiv preprint arXiv:1612.02136* (2016).

[5] Minmin Chen, Alice Zheng, and Kilian Weinberger. 2013. Fast image tagging. In *proceedings of International Conference on Machine Learning*. 1274–1282.

[6] Shang-Fu Chen, Yi-Chen Chen, Chih-Kuan Yeh, and Yu-Chiang Frank Wang. 2018. Order-free rnn with visual attention for multi-label classification. In *proceedings of AAAI Conference on Artificial Intelligence.*

[7] Pinar Duygulu, Kobus Barnard, Joao FG de Freitas, and David A Forsyth. 2002. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *proceedings of European Conference on Computer Vision.* 97–112.

[8] Weifeng Ge, Sibei Yang, and Yizhou Yu. 2018. Multi-Evidence filtering and fusion for multi-Label classification, object detection and semantic segmentation based on weakly supervised learning. In *proceedings of IEEE Computer Vision and Pattern Recognition.*

[9] Nadia Ghamrawi and Andrew McCallum. 2005. Collective multi-label classification. In *proceedings of ACM Conference on Information and Knowledge Management.* 195–200.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *proceedings of Neural Information Processing Systems.* 2672–2680.

[11] Michael Grubinger, Paul Clough, Henning Müller, and Thomas Deselaers. 2006. The IAPR TC-12 benchmark: A new evaluation resource for visual information systems. In *proceedings of OntoImage.*

[12] Matthieu Guillaumin, Thomas Mensink, Jakob Verbeek, and Cordelia Schmid. 2009. TagProp: Discriminative metric learning in nearest neighbor models for image annotation. In *proceedings of IEEE International Conference on Computer Vision.* 309–316.

[13] Baolin Guo, Chenping Hou, Feiping Nie, and Dongyun Yi. 2016. Semi-supervised multi-label dimensionality reduction. In *proceedings of IEEE International Conference on Data Mining*. 919–924.

[14] Ting Jiang, Deqing Wang, Leilei Sun, Huayi Yang, Zhengyang Zhao, and Fuzhen Zhuang. 2021. LightXML: Transformer with dynamic negative sampling for high-performance extreme multi-label text classification. *arXiv preprint arXiv:2101.03305* (2021).

[15] Feng Kang, Rong Jin, and Rahul Sukthankar. 2006. Correlated label propagation with application to multi-label learning. In *proceedings of IEEE Computer Vision and Pattern Recognition*, Vol. 2. 1719–1726.

[16] Diederik Kingma and Jimmy Ba. 2014. ADAM: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[17] Elyor Kodirov, Tao Xiang, and Shaogang Gong. 2017. Semantic autoencoder for zero-shot learning. In *proceedings of IEEE Computer Vision and Pattern Recognition*. 3174–3183.

[18] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. 2014. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 3 (2014), 453–465.

[19] Chung-Wei Lee, Wei Fang, Chih-Kuan Yeh, and Yu-Chiang Frank Wang. 2018. Multi-label zero-shot learning with structured knowledge graphs. In *proceedings of IEEE Computer Vision and Pattern Recognition*. 1576–1585.

[20] Yi Liu, Rong Jin, and Liu Yang. 2006. Semi-supervised multi-label learning by constrained non-negative matrix factorization. In *proceedings of AAAI Conference on Artificial Intelligence*, Vol. 6. 421–426.

[21] Qianqian Ma, Yang-Yu Liu, and Alex Olshevsky. 2020. Optimal lockdown for pandemic control. *arXiv preprint arXiv:2010.12923* (2020).

[22] Qianqian Ma, Yang-Yu Liu, and Alex Olshevsky. 2021. Optimal vaccine allocation for pandemic stabilization. *arXiv preprint arXiv:2109.04612* (2021).

[23] Qianqian Ma and Alex Olshevsky. 2020. Adversarial Crowdsourcing Through Robust Rank-One Matrix Completion. In *proceedings of Neural Information Processing Systems*, Vol. 33. 21841–21852.

[24] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. 2016. Least squares generative adversarial networks. *arXiv preprint arXiv:1611.04076* (2016).

[25] Andrew McCallum. 1999. Multi-label text classification with a mixture model trained by EM. In *proceedings of AAAI Conference on Artificial Intelligence*. 1–7.

[26] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

[27] Anshul Mittal, Kunal Dahiya, Sheshansh Agrawal, Deepak Saini, Sumeet Agarwal, Purushottam Kar, and Manik Varma. 2021. DECAF: Deep extreme classification with label features. In *proceedings of ACM International Conference on Web Search and Data Mining*. 49–57.

[28] Anshul Mittal, Noveen Sachdeva, Sheshansh Agrawal, Sumeet Agarwal, Purushottam Kar, and Manik Varma. 2021. ECLARE: Extreme classification with label graph correlations. In *proceedings of Web Conference*. 3721–3732.

[29] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *proceedings of International Conference on Machine Learning*.

[30] Augustus Odena, Christopher Olah, and Jonathon Shlens. 2017. Conditional image synthesis with auxiliary classifier GANs. In *Journal of Machine Learning Research*. 2642–2651.

[31] Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 10 (2010), 1345–1359.

[32] Genevieve Patterson and James Hays. 2012. SUN attribute database: Discovering, annotating, and recognizing scene attributes. In *proceedings of IEEE Computer Vision and Pattern Recognition*. 2751–2758.

[33] Guo-Jun Qi. 2020. Loss-sensitive generative adversarial networks on lipschitz densities. *International Journal of Computer Vision* 128, 5 (2020), 1118–1140.

[34] Guo-Jun Qi, Xian-Sheng Hua, Yong Rui, Jinhui Tang, Tao Mei, and Hong-Jiang Zhang. 2007. Correlative multi-label video annotation. In *proceedings of ACM International Conference on Multimedia*. 17–26.

[35] Can Qin, Lichen Wang, Qianqian Ma, Yu Yin, Huan Wang, and Yun Fu. 2021. Contradictory structure learning for semi-supervised domain adaptation. In *proceedings of SIAM International Conference on Data Mining*. 576–584.

[36] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training GANs. In *proceedings of Neural Information Processing Systems*. 2234–2242.

[37] Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *proceedings of International Conference on Learning Representations*.

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

[39] Laurens Van Der Maaten. 2014. Accelerating t-SNE using tree-based algorithms. *Journal of Machine Learning Research* 15, 1 (2014), 3221–3245.

[40] Luis Von Ahn and Laura Dabbish. 2004. Labeling images with a computer game. In *proceedings of ACM Special Interest Group on Computer-Human Interaction*. 319–326.

[41] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. 2011. *The Caltech-UCSD Birds-200-2011 Dataset.* Technical Report CNS-TR-2011-001.

[42] Lichen Wang, Zhengming Ding, and Yun Fu. 2018. Adaptive graph guided embedding for multi-label annotation.. In *proceedings of International Joint Conference on Artificial Intelligence*. 2798–2804.

[43] Lichen Wang, Zhengming Ding, and Yun Fu. 2021. Generic multi-label annotation via adaptive graph and marginalized augmentation. *ACM Transactions on Knowledge Discovery from Data* 16, 1 (2021), 1–20.

[44] Lichen Wang, Zhengming Ding, Seungju Han, Jae-Joon Han, Changkyu Choi, and Yun Fu. 2019. Generative correlation discovery network for multi-label learning. In *proceedings of IEEE International Conference on Data Mining*. 588–597.

[45] Lichen Wang, Bo Zong, Qianqian Ma, Wei Cheng, Jingchao Ni, Wenchao Yu, Yanchi Liu, Dongjin Song, Haifeng Chen, and Yun Fu. 2020. Inductive and Unsupervised Representation Learning on Graph Structured Objects. In *proceedings of International Conference on Learning Representations*.

[46] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. 2004. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612.

[47] Tong Wei and Yu-Feng Li. 2019. Does tail label help for large-scale multi-label learning? *IEEE Transactions on Neural Networks and Learning Systems* 31, 7 (2019), 2315–2324.

[48] Baoyuan Wu, Weidong Chen, Peng Sun, Wei Liu, Bernard Ghanem, and Siwei Lyu. 2018. Tagging like humans: Diverse and distinct image annotation. In *proceedings of IEEE Computer Vision and Pattern Recognition*. 7967–7975.

[49] Baoyuan Wu, Fan Jia, Wei Liu, Bernard Ghanem, and Siwei Lyu. 2018. Multi-label learning with missing labels using mixed dependency graphs. *International Journal of Computer Vision* (2018), 1–22.

[50] Baoyuan Wu, Siwei Lyu, and Bernard Ghanem. 2015. ML-MG: Multi-label learning with missing labels using a mixed graph. In *proceedings of IEEE International Conference on Computer Vision*. 4157–4165.

[51] Tong Wu, Qingqiu Huang, Ziwei Liu, Yu Wang, and Dahua Lin. 2020. Distribution-balanced loss for multi-label classification in long-tailed datasets. In *proceedings of European Conference on Computer Vision*. 162–178.

[52] Tong Wu, Ziwei Liu, Qingqiu Huang, Yu Wang, and Dahua Lin. 2021. Adversarial robustness under long-tailed distribution. In *proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 8659–8668.

[53] Xiangyun Zhao, Haoxiang Li, Xiaohui Shen, Xiaodan Liang, and Ying Wu. 2018. A Modulation Module for Multi-task Learning with Applications in Image Retrieval. In *proceedings of European Conference on Computer Vision*.

[54] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *proceedings of IEEE International Conference on Computer Vision*.

[55] Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. 2003. Semi-supervised learning using Gaussian fields and harmonic functions. In *proceedings of International Conference on Machine Learning*. 912–919.