Generic Multi-label Annotation via Adaptive Graph and Marginalized Augmentation

LICHEN WANG, Northeastern University
ZHENGMING DING, Tulane University
YUN FU, Northeastern University

Multi-label learning recovers multiple labels from a single instance. It is a more challenging task compared with single-label manner. Most multi-label learning approaches need large-scale well-labeled samples to achieve high accurate performance. However, it is expensive to build such a dataset. In this work, we propose a generic multi-label learning framework based on Adaptive Graph and Marginalized Augmentation (AGMA) in a semi-supervised scenario. Generally speaking, AGMA makes use of a small amount of labeled data associated with a lot of unlabeled data to boost the learning performance. First, an adaptive similarity graph is learned to effectively capture the intrinsic structure within the data. Second, marginalized augmentation strategy is explored to enhance the model generalization and robustness. Third, a feature-label autoencoder is further deployed to improve inferring efficiency. All the modules are jointly trained to benefit each other. State-of-the-art benchmarks in both traditional and zero-shot multi-label learning scenarios are evaluated. Experiments and ablation studies illustrate the accuracy and efficiency of our AGMA method.

CCS Concepts: • **Computing methodologies → Classification and regression trees**;

Additional Key Words and Phrases: Multi-label learning, multi-label annotation, image retrieval, adaptive graph, marginalized augmentation

ACM Reference format:
Lichen Wang, Zhengming Ding, and Yun Fu. 2021. Generic Multi-label Annotation via Adaptive Graph and Marginalized Augmentation. *ACM Trans. Knowl. Discov. Data* 16, 1, Article 12 (June 2021), 20 pages.
https://doi.org/10.1145/3451884

1 INTRODUCTION

In real-world applications, one object could relate to tens or hundreds of semantic descriptions or attributes. For instance, an image illustrates, "It is a *sunny day* with *blue sky* and a *lake/water* nearby." This image contains multiple labels (i.e., *sunny*, *blue sky*, and *water*) selected from a large number of candidate labels. Compared to single label classification task, multi-label tasks assume multiple labels exist in each instance [2, 5, 19, 21, 43]. It is a more challenging task. First, the

This research is supported in part by the U.S. Army Research Office Award (W911NF-17-1-0367).

Authors' addresses: L. Wang and Y. Fu, Northeastern University, 360 Huntington Avenue, Boston, MA 02115, USA ; emails: wanglichenxj@gmail.com, yunfu@ece.neu.edu; Z. Ding, Tulane University, 6823 St. Charles Avenue, New Orleans, LA 70118, USA; email: zding1@tulane.edu.

1556-4681/2021/06-ART12 $15.00
https://doi.org/10.1145/3451884

available datasets (e.g., SUN [40], CUB [47], and AWA [25]) are relatively small. Since multi-label data collection and labeling procedures are labor intense and expensive compared with the single-label setting. In addition, multi-label datasets suffer from high-level label noise due to the subjective nature of the labels (e.g., *hot*, *warm*, and *stressful*). It is hard to obtain consistent label results since different people hold different opinions. Third, the labels in most datasets follow a long-tail distribution. It means that some "common" labels (e.g., *blue sky*, *outdoor*, and *trees*) are much more prevalent than "rare" labels (e.g., *fight* and *fire*). For instance, the SUN dataset has $14,340$ samples in total. The most common label (i.e., *Man-made*) shows up $8,089$ times, while the rarest label (i.e., *Fire*) only shows up 73 times. The significant unbalanced training samples could negatively affect the learning performance. More sophisticated and specifically designed models are required for multi-label learning tasks.

To this end, generating such multi-label datasets is a challenging and expensive task. However, relevant and unlabeled data are easy to obtain. Based on this, semi-supervised learning [62, 63] is a practical solution to enhance the learning performance by exploring unlabeled samples. From all the various semi-supervised strategies, graph-based approaches [56] have attracted great attention due to their high performance. However, there is a major drawback. These approaches rely heavily on a high-quality similarity graph and ground-truth labels. The graph is generated based on the original sample representation which could be influenced by noise and the configurations of the similarity metrics. These factors could significantly affect the graph generation and decrease the final performance. Previous works exploited adaptive graphs to handle the noise sensitivity issue [13, 31, 32, 36, 38]. [59] proposed an error correcting output correcting scheme to achieve the multi-class heterogeneous domain adaptation. [24] learned a low-rank kernel strategy which eliminates the noise and enhances the representation ability. [22] proposed a reliable graph learning strategy. It obtains robust graphs by adaptively removing errors and noise from the original samples. [23] mapped the data into a higher dimensional space and deployed a multiple-kernel-based algorithm for recommendation system. However, most of the aforementioned approaches mainly handle the single-label classification tasks, which ignore the unique challenges of multi-label setting such as the "long-tail" label distribution issue.

Augmenting samples from the auxiliary domain is a promising direction for multi-label learning. **Marginalized Corrupted Features (MCF)** is an effective and efficient feature augmentation strategy. MCF "corrupts" existing samples and "generates" infinite artificial samples for model training [33]. It is specifically designed for the situation which only limited training samples are available. More details are introduced in [34]. [9] proposed a **marginalized Denoising Auto-encoder (mDAE)** approach for non-linear representation learning. mDAE achieves similar or even better performance with much fewer training samples. [27] proposed a Regularized Marginalized Cross-View learning framework with mDAE, which effectively improves the model robustness. However, these methods either focus on representation learning tasks or supervised classification tasks which cannot effectively explore unlabeled data.

In this article, a novel and generic multi-label learning framework via **Adaptive Graph and Marginalized Augmentation (AGMA)** strategy in semi-supervised scenario is proposed. The framework is shown in Figure 1. The core insight is jointly propagating the labeled and unlabeled data by an adaptive graph and seeking an effective and robust visual-label encoder with marginalized feature augmentation strategy. Such two strategies could assist each other to enhance the final performance. The contributions are listed below:

- An adaptive graph is proposed to explore the latent correlations of labeled and unlabeled samples. It is jointly updated with other components to obtain the best performance.

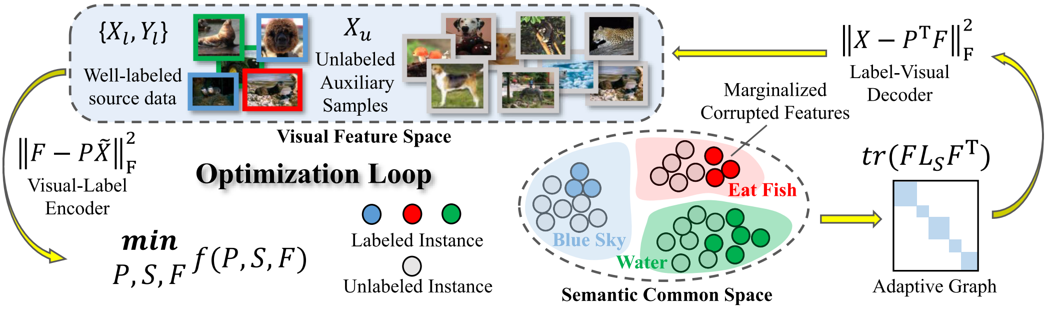


Fig. 1. Framework of AGMA approach. A visual-label encoder, P, maps data from visual space to label space. An adaptive affinity graph, S, is adaptively optimized based on both label space and feature space, which also explores the pairwise latent connections across both labeled and unlabeled data. A marginalized feature augmentation strategy is further deployed to extend the feature space and enhance the model robustness. The predicted label matrix F, the adaptive graph S, and the encoder P are jointly optimized which helps the model to obtain the best and reliable performance.

- A feature-label autoencoder is proposed to project the samples between label space and feature space. It fully explores the feature-label connection and could reduce the computational cost in the testing stage.
- A marginalized feature augmentation strategy is deployed which extends infinite samples from the limited samples and further improves the model robustness.
- An optimization approach is designed to solve all variables. Five datasets are deployed in the experiments and the results illustrate the efficiency and effectiveness of the model.

AGMA is an extension of our previous work [48]. There are three-fold modifications to improve the performance. First, we deploy an autoencoder strategy to directly project the samples between feature space and label space. It avoids the negative influence from the uncontrollable latent subspace of [48]. Second, a marginalized augmentation approach is designed to extend the feature distribution for further improving the performance. Third, our approach is efficient in inferring step, since our model is able to project the new samples from feature space to label space without extra optimization process. Extensive experiments indicate that AGMA achieves better performance. In the rest of the article, Section 2 introduces related works including semi-supervised and multi-label learning. Section 3 introduces our model. Experiments and analysis are presented in Section 4. Conclusion is provided in Section 5.

2 RELATED WORK

2.1 Multi-Label Learning

Multi-label learning predicts multiple labels from a single instance. It is a more practical and potential classification task for a large number of real-world applications, e.g., video concept recognition [42], image annotation [2], and text classification [15]. One straightforward solution for multi-label learning is utilizing multiple single-label learning classifiers to recover each label individually [2]. However, the latent correlations between labels are not considered in this strategy (e.g., *blue sky* usually show up with *outdoor*). Label relation plays an important role for multi-label learning [57]. [16] designed a contextual merging step based on the output of each classifier to leverage the correlations. [55] handles the missing label problem via learning the semantic structural information to build the label correlations. It projects samples to the semantic space with an effective semantic descriptor. [51] learned the labels as well as the correlations simultaneously in the training stage for multi-view scenario. [58] designed a dependence maximization strategy for multi-label dimension deduction based on Hilbert–Schmidt independence criterion. [8] proposed a non-negative matrix factorization to obtain robust prediction performance. [29] proposed a model which automatically identifies easy and hard prediction samples. It then uses the obtained easy samples to enhance the prediction of hard samples. However, most of these approaches are still in supervised learning manner which cannot perform well in the training data shortage situation.

2.2 Semi-Supervised Learning

Semi-supervised learning utilizes a small-scale well-labeled samples associated with a large-scale unlabeled samples to improve the learning performance [7, 36, 49, 50, 62, 63]. There are various ways to achieve semi-supervised learning. A detailed introduction can be found in [62]. Its essential insight is to explore the feature distribution knowledge from unlabeled samples and improve the effectiveness of down-stream tasks. [36] filters the training sets and obtains a model which is independent to the training initialization procedure. [60] utilized the hashing and transfer learning strategies to achieve transfer hashing for privileged information. It could handle data sparsity issues in deep learning framework. [61] proposed a self-supervised mechanism which contains two losses to achieve semi-supervised learning scenario. [20] utilizes a differentiable surrogate of the non-differentiable Hungarian algorithm to achieve the view-specific alignment. [41] effectively utilizes the knowledge from both feature and label space. The pairwise sample assignments are minimized across each data point. Graph-based approach attracts great attention due to its high accuracy and stability. It deploys an affiliate graph to explore the latent data structure residing in both source and target samples. A Gaussian random field and a harmonic function were proposed to improve the performance [63]. Although graph-based methods achieve high performance, there is a main drawback. Specifically, the classification performance heavily depends on the quality of the affiliate graph, and it is difficult to always obtain an effective affiliate graph. Moreover, most graph generation methods are parameter sensitive. Thus, the same set of graph generation configurations could not achieve the best performance for other resources. An adaptive affiliate graph is proposed in [28] which is adaptively optimized in the training stage. [35, 37] deploy graph optimization strategy for unsupervised feature selection and representation learning tasks. [52] extended this approach to image and video scenarios. [48] deployed affiliate graph associated with subspace learning to learn more distinctive feature representation and helped the adaptive graph learning. However, most of the graph-based approaches still rely on the similarity measurement in either the feature space or a learned subspace. The performance of this strategy is easily affected by noise and outliers. Moreover, these approaches ignore utilizing the latent label correlation knowledge residing inside the samples which is crucial for multi-label setting.

3 THE PROPOSED APPROACH

3.1 Preliminary

The notations utilized in this article are summarized in Table 1. Scale values or vectors are represented by lowercase letters and the matrices are illustrated by uppercase letters. $X_l \in \mathbb{R}^{d \times n_l}$ is the feature matrix of labeled data, where $X_l = [x_1, x_2, \ldots, x_{n_l}]$. d is the feature dimension, n_l is the sample number. $x_i \in \mathbb{R}^d$ represents a feature vector of the i-th sample. $Y_l \in \mathbb{R}^{d_l \times n_l}$ is the ground truth label matrix of X_l, where d_l is the label dimension. $Y_l = [y_1, y_2, \ldots, y_{n_l}]$ and $y_i \in \mathbb{R}^{d_l}$ represents a label vector. Similarly, $X_u \in \mathbb{R}^{d \times n_u}$ is the feature matrix of unlabeled data. F_l and F_u are the predicted label matrix of X_l and X_u. $F_l = [f_1, f_2, \ldots, f_{n_l}]$ and $F_u = [f_1, f_2, \ldots, f_{n_u}]$.

Table 1. Symbol Description Table

Symbol	Description
x_i, x_j	Feature vector of i-th and j-th samples.
X_l, X_u	Feature matrix of labeled and unlabeled samples.
X	$X = [X_l, X_u]$.
f_i, f_j	Predicted label vector of x_i and x_j.
F_l, F_u	Predicted multi-label of X_l and X_u, and $F = [F_l, F_u]$.
y_i, y_j	Groundtruth label vector of x_i and x_j.
Y_l	Groundtruth label of X_l.
d, d_l	Dimensions of feature space and label space.
S	Adaptive affinity graph.
L_S	Graph Laplacian matrix of S.
P	Visual-Label Encoder, and Label-Visual Decoder is $P^\top$.
n_l, n_u	Number of labeled and unlabeled samples, $n = n_l + n_u$.
λ, μ	Trade-off parameters.
δ	Gaussian distribution variance.

In semi-supervised multi-label setting, X_l, Y_l, and X_u are given. The goal of our approach is to obtain F_u as accurate as possible.

Conventional semi-supervised multi-label learning methods obtain label propagation based on a pre-defined affiliate graph [18]. This approach assumes that the pairwise samples which have high similarity scores should have similar multiple labels. In this scenario, the pre-defined affiliate graph directly determines the recovered label. However, the quality of the affiliate graph is easily affected by several aspects including different similarity metrics (e.g., Euclidean and Cosine distance), the metric configurations, and the feature/label noise. To avoid this limitation, adaptive graph-based methods are explored to automatically obtain the best graph.

Our previous work [48] learns a low-dimensional subspace to obtain distinctive representations. An adaptive affinity graph is jointly updated based on the representations. The main objective function is shown below:

$$\begin{aligned} &\min_{F,S,P} \sum_{i,j=1}^{n} \|f_i - f_j\|_2^2 s_{ij} + \mu \sum_{i,j=1}^{n} \|Px_i - Px_j\|_2^2 s_{ij}, \\ &\text{s.t. } F_l = Y_l, S \geq 0, S\mathbb{1} = \mathbb{1}, \end{aligned} \tag{1}$$

where $S \in \mathbb{R}^{n\times n}$ is the similarity matrix across all samples, each element s_{ij} is the obtained similarity score between x_i and x_j. $n = n_l + n_u$. The constraint $S\mathbb{1} = \mathbb{1}$ is included, where $\mathbb{1}$ is a vector of ones. It indicates the sum of the elements in each row is 1. This constraint controls the scale of S and avoids a trivial solution (i.e., $S = 0$). The negative influence from outliers could also be suppressed. In addition, instead of calculating the pairwise distances in the original feature space (i.e., $\|x_i - x_j\|_2^2 s_{ij}$), a linear projection $P \in \mathbb{R}^{r\times d}$ is deployed to project the original feature vectors to a low-dimensional subspace (i.e., $\|Px_i - Px_j\|_2^2 s_{ij}$). $F_l = Y_l$ since F_l is the given ground truth. F, P, and S are simultaneously optimized. By this way, S is adaptively learned based on both the feature similarity and label similarity to achieve higher prediction accuracy.

There are several drawbacks in [48] may still limit its potential performance. First, it is difficult to guarantee that the learned subspace can obtain the most distinctive representations. High level noise could reduce the quality of the subspace. Second, the approach does not well solve the limited training data challenge. Third, if new/unseen samples should be predicted in the testing stage, the whole optimization procedure has to be operated again to obtain the prediction result, F. To this end, this pipeline is not efficient for large-scale applications.

3.2 Visual-Label Encoder via Adaptive Graph

To solve the aforementioned drawbacks, we improve the model by directly projecting the data from feature space to label space. In our new model, a projection P is trained to output the label prediction as shown below:

$$f_i = Px_i. \tag{2}$$

By this way, the connection between features and labels could be further tightened and it avoids the potential negative influence from the arbitrary subspace. Furthermore, we assume the predicted label vector could still recover the original features, inspired by the work of semantic autoencoder [14], we let the encoder share the same weight as P. This strategy could further help the model to reduce the computational cost and mitigate overfitting. To this end, we have

$$x_i = P^\top f_i. \tag{3}$$

By replacing the second term in Equation (1) with Equations (2) and (3), we can have the objective function shown below:

$$\begin{aligned} &\min_{F,P,S} \sum_{i,j=1}^{n} \|f_i - f_j\|_2^2 s_{ij} + \mu \sum_{i=1}^{n} \|f_i - Px_i\|_2^2 + \lambda \sum_{i=1}^{n} \|x_i - P^\top f_i\|_2^2, \\ &\text{s.t. } F_l = Y_l, S \geq 0, S\mathbb{1} = \mathbb{1}, \end{aligned} \tag{4}$$

where P projects visual feature to the label/semantic space and $P^\top$ maps the predicted labels back to the original feature space. The second and the third term calculate the encoder error and decoder error respectively. λ and μ are the trade-off parameters which balance the weight between label space and visual space. S is initialized as a dense matrix in the optimization process. It gradually converged to a sparse matrix due to the constraint $S\mathbb{1} = \mathbb{1}$. The sparsity of S is influenced by the data distribution of different datasets.

To make Equation (4) more compact and efficient to solve, we rewrite Equation (4) as a matrix format which is shown below:

$$\begin{aligned} &\min_{F,P,S} \operatorname{tr}\,(FL_S F^\top) + \mu\|F - PX\|_\text{F}^2 + \lambda\|X - P^\top F\|_\text{F}^2, \\ &\text{s.t. } F_l = Y_l, S \geq 0, S\mathbb{1} = \mathbb{1}, \end{aligned} \tag{5}$$

where tr(.) indicates the matrix trace calculation which is the sum of the main diagonal elements. $L_S \in \mathbb{R}^{n\times n}$ is the Laplacian matrix. $L_S = D - S$ where $D \in \mathbb{R}^{n\times n}$ and $D_{ii} = \sum_{i=1}^{n} s_{ij}$. $X = [X_l, X_u]$ and $F = [F_l, F_u]$.

3.3 Generic Encoder Learning via Marginalized Augmentation

Long-tail label distribution is common in multi-label learning, which means some labels only have very limited training samples. This challenge also suppresses the learning performance. To address this problem, we explore the idea of MCF [33]. It effectively extends/enlarges the feature distribution by corrupting the existing training examples with a fixed noise distribution. By this way, the feature distribution gaps between samples could be filled up.

Given a feature vector $x_i \in \mathbb{R}^d$. We let x_i^k ($k = \{1, 2, \dots, d\}$) represent the value of each dimension of x_i. MCF assumes that the augmentation distribution factorizes over all dimensions of x_i. It considers each individual distribution as a combination of a set of natural exponential family:

$$p(\widetilde{x}_i|x_i) = \prod_{k=1}^{d} P_E(\widetilde{x}_i^k|x_i^k; \eta_k), \tag{6}$$

where $\widetilde{x}_i$ is the corrupted version of x_i. η_k is the augmentation distribution parameter on the dimension k. MCF constrains $\mathbb{E}[\widetilde{x}_i]_{p(\widetilde{x}_i|x_i)} = x_i$, where $\mathbb{E}(\widetilde{x}_i)$ is the expectation of $\widetilde{x}_i$. It means that the expectation of the augmented features should be the same as x_i.

In our model, all the samples from labeled and unlabeled sets are utilized to obtain the corrupted features. Given the whole samples $\mathcal{D} = [(x_i, f_i)]_{i=1}^{n}$, assume we augment the samples M times and obtain the augmented features $\widetilde{x}_{im}$ ($m = 1, 2, 3, \ldots, M$). Then, our model can utilize these features $\widetilde{\mathcal{D}}$ to train any classification models by minimizing the equation below:

$$\mathcal{L}(\mathcal{D}; \Theta) = \sum_{i=1}^{n} \frac{1}{M} \sum_{m=1}^{M} L(\widetilde{x}_{im}, f_i; \Theta), \tag{7}$$

where Θ is the model parameters with $\widetilde{x}_{im} \sim p(\widetilde{x}_{im}|x_i)$, and $L(x_i, f_i; \Theta)$ is the objective function of a proposed model. However, such approach is not elegant and could increase the computational cost significantly. To this end, the limiting case in which $M \to \infty$ can be used for Equation (7) as follows:

$$\mathcal{L}(\mathcal{D}; \Theta) = \sum_{i=1}^{n} \mathbb{E}[L(\widetilde{x}_i, y_i; \Theta)]_{p(\widetilde{x}_i|x_i)}, \tag{8}$$

where $\mathbb{E}(.)$ is the expectation of the objective value. Minimizing Equation (8) under the corruption model is the crucial module for MCF. The solution of Equation (8) relies heavily on the objective function and the augmentation distributions. Coincidentally, for projections that employ exponential or quadratic objective function, the expectations in Equation (8) could be obtained for all augmentation distributions in the natural exponential family [33]. To this end, we modify Equation (5) based on the MCF strategy and the expression can be formulated as follows:

$$\begin{aligned} &\min_{F,P,S} \mathrm{tr}(FL_SF^\top) + \mu\mathbb{E}\left[\|F - P\tilde{X}\|_\mathrm{F}^2\right] + \lambda\|\tilde{X} - P^\top F\|_\mathrm{F}^2, \\ &\text{s.t. } F_l = Y_l, S \geq 0, S\mathbb{1} = \mathbb{1}, \end{aligned} \tag{9}$$

where $\tilde{X}$ is the corrupted features of X. We preserve the quadratic objective loss and deploy the isotropic Gaussian distribution to augment the feature with mean x_i and variance $\delta^2\mathbf{I}$. In this way, the expectation can be written as a simple case as follows:

$$\begin{aligned} \mathbb{E}\left[\|F - P\tilde{X}\|_\mathrm{F}^2\right] &= P(\mathbb{E}[\tilde{X}]\mathbb{E}[\tilde{X}]^\top + V[\tilde{X}])P^\top - 2\mathrm{tr}(Y\mathbb{E}[\tilde{X}])^\top P^\top + \mathrm{tr}(FF^\top), \\ &= P\mathrm{tr}(XX^\top)P^\top - 2(YX)^\top P^\top + \delta^2 nPP^\top + \mathrm{tr}(FF^\top), \end{aligned} \tag{10}$$

where $V[\tilde{X}]$ is a diagonal matrix storing the variance of X. It is the standard l_2-regularized quadratic objective function. Combined with other terms, Equation (7) can be shown as follows:

$$\begin{aligned} &\min_{F,P,S} \mathrm{tr}(FL_SF^\top) + \mu\mathrm{tr}(PXX^\top P^\top) - 2\mu\mathrm{tr}(FX^\top P^\top) + \mu\mathrm{tr}(\delta^2 nPP^\top + FF^\top) + \lambda\|X - P^\top F\|_\mathrm{F}^2, \\ &\text{s.t. } F_l = Y_l, S \geq 0, S\mathbb{1} = \mathbb{1}. \end{aligned} \tag{11}$$

Equation (11) is the complete objective function of our model. Deploying MCF does not increase the computational cost significantly since the complexity of the training algorithms remains linear in n. Our model is easy to degrade to non-augmented version. From Equation (11), we observe that Equation (11) becomes exactly the same as Equation (5) when $\delta = 0$. We will further prove the effectiveness of marginalization augmentation by tuning the value of δ.

Compared with our previous work, AG^{2}E [48], our approach has an extra advantage. When new/unseen samples come, our model could directly infer their labels by $F_{new} = PX_{new}$, where X_{new} are the new samples and F_{new} are the predicted labels. Such a strategy avoids the optimization procedure. Although the feature distribution knowledge of new data could not be fully explored, it is an effective and efficient way and the performance is still high and stable since P is well trained. More theoretical analysis is provided in Section 3.5, and the empirical evaluation is shown in Section 4.9.

Our approach is able to handle domain shift issue between labeled and unlabeled samples, which is similar to domain adaptation approaches. While, there are several differences between them. Conventional domain adaptation approaches explicitly learn the domain-invariant representation, while our approach achieves domain adaptation by exploring the sample similarities across different domains and adjusting the similarity matrix. Moreover, conventional methods mainly diminish the domain shift only in feature space, while our approach adaptively explores the similarities in both feature space and label space.

ALGORITHM 1: *Solution to Equation* (11)

```
Input:
  labeled and unlabeled feature matrices $X_l$ and $X_u$,
  label matrix of $Y_l$, Gaussian distribution variance $\delta$,
  trade off parameter $\mu$, $\lambda$ and convergent threshold $\epsilon$.
Output:
  The recovered label $F_u$, semantic projection $P$.
Initialization:
  Train $\min_P \|Y_l - P^\top X_l\|_F^2 + \mu\|P\|_F^2$ and initial $F_u = P^\top X_u$,
  Obtain $F$ by concatenating $Y_l$ and $F = [Y_l, F_u]$.
Optimization:
while not converged do
    Update $P_{(k+1)}$ from the solution of (11);
    while not converged do
        Update $S_{i(k+1)}$ using Equation (11);
    end while
    Calculate $L_s = D_s - (S + S^\top)/2$, $D_{sii} = \sum_i (S_{ij} + S_{ji})/2$;
    Update $F_u$ using Equation~(11), given others fixed;
    $k = k + 1$;
    Obtain $\mathcal{L}_k$, which is the objective value of Equation~(11)
    Check if $|\mathcal{L}_{k-1} - \mathcal{L}_k| < \epsilon$.
end while
```

3.4 Optimization

Three variables in Equation (11) are required to be optimized. It is difficult to obtain an explicit solution. We adopt the **Alternative Directions Method of Multipliers (ADMM)** [3] to solve the problem. ADMM is driven by alternatively optimizing the equation with respect to P, S, and F. The pseudocode of the optimization procedure is provided in Algorithm 1. P_0 is the initialization of P, it is initialized based on the objective function $\min_{P_0} \|Y_l - P_0 X_l\|_F^2 + \mu_0 \|P_0\|_F^2$, where μ_0 is a trade-off parameter and empirically set to 100. Then F_u is initialized by $F_u = PX_u$. After that, ADMM is deployed to update one variable each time where other variables are fixed. All the variables are iteratively optimized until Equation (11) is convergent. We introduce the details of the optimization procedure below:

Update P: When others are fixed, Equation (11) can be written as below:

$$\min_P \mathrm{tr}(PXX^\top P^\top) - 2\mathrm{tr}[(FX^\top)P^\top] + \mathrm{tr}\left[\delta^2 nPP^\top + FF^\top\right] + \tfrac{\lambda}{\mu}\|X - P^\top F\|_F^2. \tag{12}$$

To obtain the optimized point, we assign the derivation of Equation (12) with respect of P to zero and obtain:

$$2PXX^\top - 2\left[(FX^\top)\right] + 2\delta^2 nP + \tfrac{2\lambda}{\mu}F(F^\top P - X^\top) = 0, \tag{13}$$

then Equation (13) can be simplified to the following equation:

$$(\delta^2 nI + \frac{\lambda}{\mu}FF^\top)P + P(XX^\top) = (1 + \frac{\lambda}{\mu})FX^\top. \tag{14}$$

Since Equation (14) is a Sylvester equation, the Bartels–Stewart algorithm [1] can be deployed to efficiently solve the equation.

Update S: By ignoring other variables, Equation (11) can be written as below:

$$\begin{aligned}&\min_S \mathrm{Tr}(FL_S F^\top),\\ &\text{s.t.}\ \ S \geq 0, S\mathbb{1} = \mathbb{1}.\end{aligned} \tag{15}$$

S cannot be explicitly solved due to the two constraints $S \geq 0$ and $S\mathbb{1} = \mathbb{1}$. We optimize S row by row, based on this strategy, the equation can be written as follows:

$$\min_S \sum_{i=1}^{n} \|f_i - f_j\|_2^2 s_{ij} = \sum_{i=1}^{n} a_i \mathbf{s}_i^\top, \tag{16}$$

where $a_i = \{a_{ij}, 1 \leq j \leq n\} \in \mathbb{R}^{1\times n}$ with $a_{ij} = \|f_i - f_j\|_2^2$, $\mathbf{s}_i$ is the i-th row of S. KKT [4] approach can be used for solving this problem, then the updated graph S is obtained.

Update F: When others are fixed, the objective function can be written as follows:

$$\begin{aligned}&\min_F \mathrm{tr}(FL_s F^\top) - 2\mu\mathrm{tr}\left[(FX^\top)P^\top\right] + \mu\mathrm{tr}(FF^\top) + \lambda\|X - P^\top F\|_\mathrm{F}^2,\\ &\text{s.t.}\ \ F_l = Y_l.\end{aligned} \tag{17}$$

Since label matrix F is the concatenation of labeled and unlabeled data (i.e., $F = [F_l, F_u]$), thus, we can decompose Equation (17) and obtain the equation shown below:

$$\begin{aligned}&\min_{F_u} \mathrm{tr}([F_l, F_u]L_s[F_l, F_u]^\top) - 2\mu\mathrm{tr}([F_l, F_u]X^\top)P^\top\\ &\qquad + \mu\mathrm{tr}([F_l, F_u][F_l, F_u]^\top) + \lambda\|X - P^\top[F_l, F_u]\|_\mathrm{F}^2,\\ &\text{s.t.}\ \ F_l = Y_l.\end{aligned} \tag{18}$$

Meanwhile, L_s can also be decomposed as $L_s = \begin{bmatrix} L_{ll} & L_{lu} \\ L_{ul} & L_{uu}\end{bmatrix}$. Then, Equation (18) can be further decomposed as shown below:

$$\begin{aligned}&\min_{F_u} \mathrm{tr}\left(F_l L_{ll} F_l^\top + F_u L_{ul} F_l^\top + F_l L_{lu} F_u^\top + F_u L_{uu} F_u^\top\right) - 2\mu\mathrm{tr}\left[\left(F_l X_l^\top + F_u X_u^\top\right)P^\top\right]\\ &\qquad + \mu\mathrm{tr}\left(F_l F_l^\top + F_u F_u^\top\right) + \lambda\|X_u - P^\top F_u\|_\mathrm{F}^2,\\ &\text{s.t.}\ \ F_l = Y_l.\end{aligned} \tag{19}$$

To obtain the optimized point, we assign the derivation of Equation (19) with respect of F_u to zero and obtain:

$$\left(L_{ul}F_l^\top\right)^\top + F_l L_{lu} + F_u L_{uu} + F_u L_{uu}^\top - 2\mu P X_u + 2\mu F_u + 2\lambda P(P^\top F_u - X_u) = 0. \tag{20}$$

By simplifying Equation (20), Bartels–Stewart algorithm [1] can be used to solve the equation:

$$(\mu I + \lambda PP^\top)F_u + F_u L_{uu} = (\mu + \lambda)PX_u - F_l L_{lu}. \tag{21}$$

We set a threshold ϵ, if the difference is less than ϵ, then we consider the optimization process is converged. Then we stop the process and report the final performance.

Table 2. Datasets Statistical Summary

Datasets	Setting	Labeled	Unlabeled	Labels	Ave
SUN [40]	General	6,387	6,513	102	6.3
	Zero-shot	12,900	1,440		
CUB [47]	General	4,374	4,468	312	31.4
	Zero-shot	8,842	2,946		
AWA [26]	General	12,154	12,141	85	15.0
	Zero-shot	24,295	6,180		
BIRD [6]	General	322	323	19	1.1
EMO [46]	General	391	202	6	1.9

3.5 Complexity Analysis

In the optimization stage, updating P and F requires the Bartels–Stewart approach and the complexity becomes $\mathbf{O}(d^3)$ and $\mathbf{O}(n^3)$, respectively. These steps have more efficient solution by Coppersmith–Winograd algorithm [11] and the computational cost can be reduced to $\mathbf{O}(d^{2.37})$ and $\mathbf{O}(n^{2.37})$. To this end, the sum of the complexity is $\mathbf{O}(td^{2.37} + tn^{2.37})$ where t is the iteration number. The obtained computational cost is the cost for the whole optimization procedure. It could fully explore the data structure from both labeled and unlabeled samples. However, as mentioned in Section 3.3, we can utilize the learned projection P to directly infer the new/unseen samples (i.e., $F_{new} = PX_{new}$). This strategy avoids the optimization procedure which is more efficient. By this way, we reduce the complexity to $\mathbf{O}(n)$. It is more suitable for large-scale real-world applications.

4 EXPERIMENTS

To comprehensively evaluate the effectiveness of our approach, we tested our AGMA as well as other baselines in both general and zero-shot multi-label learning scenario. Zero-shot setting is more challenging which attempts to recover labels from the "unseen" samples. The details will be introduced in Section 4.4.

4.1 Datasets

Five multi-label datasets including one emotion dataset, one acoustic dataset, and three image datasets. Brief introductions are listed as follows, and the statistical summary of the datasets is listed in Table 2.

SUN Dataset [40] is widely used in fine-grained scene understanding and high-level scene recognition. It contains $14,000$ samples collected from 700 classes. Each sample has a 102-dimensional label vector which contains averagely 6.3 labels. The label value is in $\{0, 0.33, 0.66, 1\}$, since there are three annotators label each image, and the dataset averages the assigned label from all the annotators.

CUB Dataset [47] is an augmentation dataset derived from CUB-200 dataset [53]. It contains 200 categories of birds. There are 312 attribute label candidates. The elements in the label vector are binary values, i.e., 0 and 1.

AWA Dataset [26] is a large-scale animal attribute datasets, where more than 30,000 samples are collected from 50 animal categories. The label is a 85-dimensional vector with the continuous element values from 0 to 100. There are around 15 labels of each sample.

BIRD Dataset [6] contains the acoustic recordings collected from 19 different kinds of birds. Each recording is around 10 seconds in length. The recordings are paired with its attributes assigned by several experts along with their confidence. Each label vector contains binary value in $\{0, 1\}$.

EMO Dataset [46] captures the music from 233 musical albums. It aims to test the music emotion evaluation approaches. There are 593 songs where each song is extracted to a 30-seconds recording and classified to 6 emotions assigned by music experts.

For the image datasets (i.e., SUN, CUB, and AWA datasets), Very Deep Convolution Networks [44] pre-trained by ImageNet [12] is utilized to extract deep features. It obtains the $4,096$-dimensional feature vector for each instance. We also evaluate GoogleNet [45] features on these datasets and observe that different features may cause different performances, while our approach always achieves high performance. For the BIRD dataset, we use the features provided by [6]. Both the Rhythmic and the Timbre features provided by [46] are utilized for the EMO dataset.

4.2 Experimental Setup

Traditional multi-label scenario and the zero-shot multi-label scenario [25, 39] settings are deployed in our experiments. In the conventional setting, we randomly extract the samples from the whole datasets and build a labeled set and an unlabeled set. Each set has half of the whole sample. Our model is evaluated five times based on the randomly generated training/testing sets and report the average performance. The standard deviation is also provided. Five-fold cross-validation is deployed to tune the trade-off parameters λ and μ. The parameter sensitivity analysis will be introduced in the experiments. We evaluate our methods as well as other state-of-the-art multi-label learning methods. The brief introduction of all the baselines are shown below:

- **Least Squares Regression (Regression)** is a ridge regression approach. It obtains a projection based on the training samples and then recovers the target samples.
- **Semi-Supervised Multi-Label Dimensionality Reduction (SSMLDR)** [18] enlarges the multiple label information from the labeled samples to the unlabeled samples. In addition, a transformation matrix is proposed to obtain the distinctive low-dimensional representations.
- **FastTag** [10] proposes two linear projections that are simultaneously optimized in a joint convex objective function. Even if the training samples contain incomplete/noisy ground truth labels, FastTag is able to effectively and efficiently predict the complete list of labels.
- **Multi-Label with a Mixed Graph (ML-PGD)** [54] designs a mixed graph which fully explores the label dependencies. It considers the co-occurrence across each pair of the candidate labels and the instance-level similarities as the graph edges.
- **Semantic AutoEncoder (SAE)** [14] proposes an effective and efficient autoencoder strategy. It recovers multiple labels without other sophisticated constraints. SAE achieves high performance in both conventional and zero-shot learning settings.
- **Adaptive Graph Guided Embedding (AG^{2}E)** [48] proposes a novel approach which simultaneously updates the affinity graph, recovers labels, and optimizes projected subspaces. It effectively overcomes the label noise and long-tail distribution issues.

We deploy the metrics utilized in [17]. Specifically, the recall R and the precision (Prec) P are obtained. $P = \frac{t_p}{t_p+f_p}$ and $R = \frac{t_p}{t_p+f_n}$, where t_p denotes true-positive. f_n and f_p represent the false-negative and the false-positive, respectively. We calculate harmonic mean of the precision and the recall, F1-score (F1), to compare the results easier. $F_1 = 2\frac{P\times R}{P+R}$. A **non-zero recall (N-R)** which denotes the number of non-zero labels are further reported. Moreover, the **mean average precision (mAP)** utilized in [54] is further deployed for a comprehensive evaluation. For all evaluations, higher value denotes better performance.

Table 3. Performance Comparison with Other Methods

Dataset	Method	Prec	Recall	F1	N-R	mAP
SUN	Regression	0.6318±0.0070	0.1504±0.0011	0.2429±0.0016	100.0±0.0000	0.3907±0.0026
	SSMLDR	0.5625±0.0021	0.1239±0.0011	0.2031±0.0045	67.8±2.0736	0.6315±0.0038
	FastTag	0.6187±0.0251	0.1473±0.0027	0.2379±0.0083	101.0±0.4265	0.6935±0.0189
	ML-PGD	0.3218±0.0178	0.1521±0.0009	0.2513±0.0010	100.2±0.3235	0.7013±0.0016
	SAE	0.7415±0.0089	0.1976±0.0005	0.3123±0.0011	101.4±0.5477	0.6928±0.0019
	AG^{2}E	**0.7460±0.0063**	0.1625±0.0019	0.2669±0.0028	102.0±0.0000	**0.7174±0.0013**
	Ours	0.7046±0.0144	**0.2040±0.0015**	**0.3164±0.0018**	**102.0±0.0000**	0.6821±0.0028
CUB	Regression	0.2728±0.0080	0.0317±0.0007	0.0568±0.0013	166.6±1.7889	0.2831±0.0035
	SSMLDR	0.2162±0.0031	0.0399±0.0003	0.0674±0.0006	163.8±2.8636	0.2135±0.0033
	FastTag	0.3231±0.0244	0.0496±0.0028	0.0860±0.0052	163.0±4.2426	0.2457±0.0255
	ML-PGD	0.3029±0.0067	0.0448±0.0002	0.0781±0.0004	132.4±3.1937	0.4081±0.0049
	SAE	0.2947±0.0062	0.0424±0.0007	0.0742±0.0014	175.6±5.4498	0.4020±0.0027
	AG^{2}E	0.3351± 0.0079	0.0525±0.0009	0.0908±0.0015	194.2±3.1195	0.4011±0.0027
	Ours	**0.3976±0.0048**	**0.0578±0.0007**	**0.1010±0.0009**	**200.4±1.1670**	**0.4115±0.0046**
AWA	Regression	0.8198±0.0098	0.0819±0.0001	0.1489±0.0003	74.8±0.8366	0.9282±0.0003
	SSMLDR	0.8085±0.0087	0.0948±0.0002	0.1698±0.0004	74.0±0.8366	0.8323±0.0031
	FastTag	0.7848±0.0316	0.0857±0.0031	0.1545±0.0096	67.2±3.1852	0.8851±0.0183
	ML-PGD	0.5283±0.0019	0.0631±0.0001	0.1127±0.0004	44.6±1.6733	0.9103±0.0001
	SAE	**0.9506±0.0010**	0.1029±0.0005	**0.1857±0.0007**	75.2±0.8944	0.8630±0.0001
	AG^{2}E	0.7745±0.0096	**0.1285±0.0016**	0.2204±0.0027	71.8±1.0062	0.9211±0.0074
	Ours	0.9013±0.0092	0.0971±0.0018	0.1766±0.0030	**81.0±0.4472**	**0.9355±0.0073**
EMO	Regression	0.3793±0.0053	0.9114±0.0118	0.5357±0.0069	6.0±0.0000	0.5431±0.0127
	SSMLDR	0.3556±0.0048	0.8965±0.0094	0.5093±0.0078	6.0±0.0000	0.5590±0.0103
	FastTag	0.3833±0.0198	0.9459±0.0215	0.5456±0.0272	6.0±0.0000	0.5894±0.0428
	ML-PGD	0.3784±0.0079	0.9265±0.0078	0.5373±0.0090	6.0±0.0000	0.5677±0.0135
	SAE	0.3923±0.0143	0.8389±0.0083	0.5346±0.0157	6.0±0.0000	0.5770±0.0153
	AG^{2}E	0.3995±0.0122	**0.9714±0.0131**	0.5762±0.0121	6.0±0.0000	0.5825±0.0181
	Ours	**0.4474±0.0080**	0.8361±0.0230	**0.5829±0.0118**	**6.0±0.0000**	**0.5962±0.0201**
BIRD	Regression	0.0764±0.0078	0.3726±0.0367	0.1268±0.0128	12.8±0.7071	0.2364±0.0546
	SSMLDR	0.0709±0.0052	0.3465±0.0282	0.1178±0.0093	12.2±0.7071	0.1436±0.0382
	FastTag	0.1005±0.0144	0.3783±0.0421	0.1601±0.0153	15.6±1.1400	0.1643±0.0857
	ML-PGD	0.0809±0.0089	0.3883±0.0267	0.1338±0.0134	15.4±1.0000	0.2423±0.0329
	SAE	0.0964±0.0107	0.3665±0.0435	0.1526±0.0156	15.2±1.3038	0.1779±0.0480
	AG^{2}E	0.1021±0.0150	0.4529±0.0186	0.1653±0.0187	16.8±0.7786	0.2454±0.0466
	Ours	**0.1065±0.0131**	**0.5216±0.0181**	**0.1780±0.0143**	**18.0±0.0000**	**0.3519±0.0311**

4.3 Performance Comparison

Table 3 shows the classification evaluations. The result illustrates the higher performance is obtained by our approach than other methods in most of the metrics. In addition, we can see that the deviations of all the evaluated methods are relatively low. Although the deviations of our approach are not the smallest, it is small enough to demonstrate the significance and stability of our method.

We observe that the mAP performance is not competitive in the AWA dataset. We conjecture several reasons. First, in the AWA dataset, the samples which belong to the same class have consistent label vectors. Consider there are only 50 different label vectors corresponding to the 50 classes. The label distribution/diversity is narrow and this situation is unique in the AWA dataset.

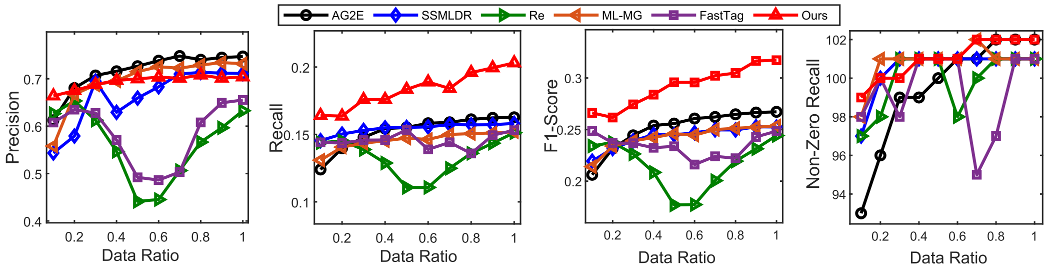


Fig. 2. Multi-label annotation performance based on part of the available training samples. It denotes that our approach still achieves high performance when fewer and fewer labeled samples are provided in the training stage. It demonstrates the effectiveness and stability of the trained model.

We assume it is hard for our approach to learn the comprehensive distribution knowledge and augment diverse features. Second, the AWA dataset contains 24, 295 samples with averagely 15 labels in each sample. The dataset scale is bigger than other datasets. We conjecture that the data scale is already big enough for training a good classifier, and our model gains limited benefits from the feature augmentation strategy. Meanwhile, our model still gets the best performance in mAP metric which is considered as one of the most important metrics (i.e., F1 and mAP) for multi-label learning scenario. For the SUN dataset, we observed that the precision and mAP are not the highest performance. We assume that although the feature augmentation strategy is effective for improving the performance, the precision-recall improvement balances of different datasets are uncertain. We observe that in most of the cases either precision or recall is higher than other state-of-the-art methods. F1 metric is a comprehensive evaluation which considers both precision and recall, and our method obtains the highest performance in most of the target datasets.

4.4 Zero-shot Multi-label Classification

More challenging zero-shot scenario is deployed for evaluating our approach. In zero-shot setting, the classes in the training set and the test set have no overlap, which means the feature distribution gaps between training and test sets are more significant. Specifically, in multi-label scenario, all the samples share the same set of multi-label candidates, while the training and test samples are extracted from non-overlapped categories (e.g., *horse* and *zebra* could be in training and test sets, respectively. They share similar shape labels but different color/texture labels). SUN, CUB, and AWA datasets have the default splits for zero-shot scenario. Specifically, in SUN dataset, it contains 645 training classes and 72 test classes. In CUB dataset, 150 bird categories are used for training and the rest 50 categories are used for testing. Moreover, AWA dataset consists 40 training classes and 10 test classes. The detailed sample numbers are further summarized in Table 2.

The same evaluation metrics as a general multi-label task are deployed and the results are illustrated in Table 4. We can observe that our approach achieves higher performance compared with other baselines. The result illustrates the ability of our approach for handling domain shift scenario. The standard deviations are still small while slightly higher than conventional multi-label setting. We assume it is due to the larger distribution gap across training and test data in zero-shot scenario.

4.5 Model Robustness Analysis

To estimate the robustness of our model, we use only partial samples from the labeled set (from 10% to 100%) and the final results are shown in Figure 2. From Figure 2, we observe that our approach

Table 4. Zero-shot Multi-label Learning Performance

Dataset	Method	Prec	Recall	F1	N-R	mAP
SUN	Regression	0.4301±0.0083	0.1243±0.0018	0.1929±0.0023	**62.0±0.0000**	0.4142±0.0035
	SSMLDR	0.2611±0.0029	0.1055±0.0018	0.1503±0.0061	48.2±2.2893	0.3516±0.0046
	FastTag	0.3924±0.0316	0.1317±0.0042	0.1972±0.0152	60.6±3.1825	0.3775±0.0227
	ML-PGD	0.2972±0.0198	0.1138±0.0013	0.1646±0.0020	34.6±2.5273	0.5181±0.0025
	SAE	0.4838±0.0128	0.1210±0.0007	0.1943±0.0015	55.8±0.6285	**0.5357±0.0021**
	AG^{2}E	**0.4925±0.0059**	0.1235±0.0028	0.1975±0.0041	55.2±0.1685	0.5132±0.0017
	Ours	0.4710±0.0162	**0.1326±0.0017**	**0.2069±0.0020**	57.8±2.2114	0.4739±0.0031
CUB	Regression	0.2026±0.0091	0.0268±0.0009	0.0474±0.0018	143.6±1.9128	0.1982±0.0044
	SSMLDR	0.1949±0.0042	0.0360±0.0004	0.0607±0.0008	131.4±3.1010	0.2535±0.0038
	FastTag	0.2821±0.0286	0.0428±0.0033	0.0743±0.0074	143.0±3.6278	0.2229±0.0266
	ML-PGD	0.1953±0.0081	0.0357±0.0002	0.0604±0.0006	81.8±2.4681	0.3095±0.0061
	SAE	0.2206±0.0083	0.0355±0.0009	0.0611±0.0019	138.4±5.1826	0.3064±0.0035
	AG^{2}E	0.2749±0.0086	0.0415±0.0011	0.0720±0.0017	172.0±2.1983	**0.3115±0.0036**
	Ours	**0.2838±0.0062**	**0.0446±0.0009**	**0.0768±0.0011**	**172.2±1.8315**	0.3004±0.0050
AWA	Regression	0.7761±0.0151	0.0761±0.0004	0.1386±0.0007	68.4±1.0425	0.8818±0.0012
	SSMLDR	0.7380±0.0121	0.0787±0.0003	0.1423±0.0004	67.6±1.2185	0.8423±0.0082
	FastTag	0.7753±0.0451	0.0852±0.0052	0.1535±0.0165	65.8±3.8195	**0.8838±0.0267**
	ML-PGD	0.4570±0.0026	0.0607±0.0002	0.1073±0.0005	39.8±2.1066	0.8431±0.0004
	SAE	0.8914±0.0016	**0.0920±0.0007**	0.1648±0.0011	71.6±1.1528	0.8432±0.0004
	AG^{2}E	0.8810±0.0132	0.0897±0.0018	0.1511±0.0035	71.8±1.1225	0.8381±0.0093
	Ours	**0.9129±0.0129**	0.0906±0.0028	**0.1657±0.0052**	**84.0±0.6385**	0.8493±0.0085

Table 5. Ablation Study of Marginalized Augmentation Strategy

Dataset	Aug.	Prec	Recall	F1	N-R	mAP
EMO	×	0.4215±0.0071	0.8357±0.0294	0.5611±0.0152	5.0±0.0000	0.5832±0.0187
	√	**0.4474±0.0080**	0.8361±0.0230	**0.5829±0.0118**	**6.0±0.0000**	**0.5962±0.0201**
BIRD	×	0.1051±0.0189	0.5113±0.0201	0.1735±0.0113	17.0±0.0000	0.3391±0.0253
	√	**0.1065±0.0131**	**0.5216±0.0181**	**0.1780±0.0143**	**18.0±0.0000**	**0.3519±0.0311**
SUN	×	0.6953±0.0096	0.1914±0.0024	0.3011±0.0031	100.0±0.0000	0.6785±0.0030
	√	**0.7046±0.0144**	**0.2040±0.0015**	**0.3164±0.0018**	**102.0±0.0000**	**0.6821±0.0028**

is still able to secure the high performance even only 20% labeled samples are provided, and it achieves the highest performances in most of the metrics when the ratio is from 20% to 100%. The results prove the robustness of our model with limited samples.

4.6 Marginalized Feature Augmentation

To demonstrate the effectiveness of the marginalized augmentation strategy, we evaluated the performance with and without augmentation module. As we discussed in Section 3.3, our model can degrade to a non-augmentation version when the variation of the augmented feature distribution, δ, is reduced to zero. To this end, we tested the performance with and without it by tuning $\delta = 0$, and the result is shown in Table 5. Moreover, we gradually increase δ value and report the performance. The results in the BIRD dataset are illustrated in Figure 3. From the results, we observe that as δ increases, almost all the metrics have some improvements. This result demonstrates

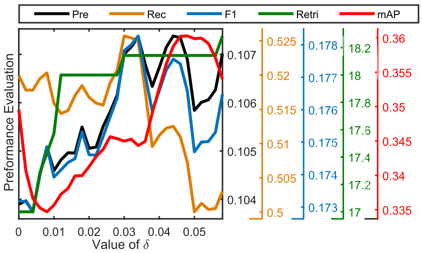


Fig. 3. Learning performance based on different values of the Gaussian distribution variance δ for feature augmentation. Different colors indicate the five metrics respectively. The result shows that almost all the metrics improve as δ increases. It demonstrates the effectiveness of MCF module.

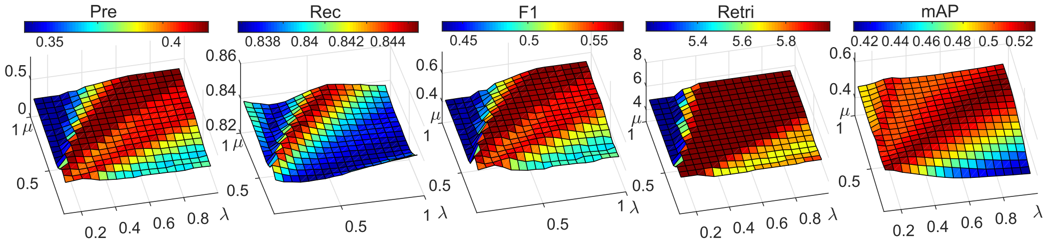


Fig. 4. Parameter sensitivity analysis of μ and λ. The much "redder" of the color indicates the higher of the performance, and vice versa. From the results we observe that there are a wide range of values which could make our model achieve the best performance. It proves the effectiveness and robustness of our model. In real-world applications, cross-validation can be utilized for parameter tuning.

the effectiveness of the marginalized augmentation for improving the performance. In the experiments, we notice that the same type of feature achieves the highest performance based on the same δ, and different features require different δ. We utilize cross-validation to tune δ and report the performances. In addition, we observe that the performances of different values of δ are relatively independent to other variables (i.e., μ and λ). Therefore, we tune δ after other parameters are tuned. It is a more practical strategy in real-world applications.

4.7 Model Analysis

We further visualize the performance based on different values of μ and λ to analyze the parameter sensitivity. The result is shown in Figure 4. The color scale bar from blue to red indicates the performance from low to high. From Figure 4, we can obtain two conclusions. First, both μ and λ could affect the performance. Second, there is a large region (i.e., red region) in the visualization result where μ and λ are roughly equal to each other. This configuration usually leads to the best performance. In our parameter tuning process, we usually set one parameter fixed (e.g., $\mu = 1$) and utilize cross-validation strategy to tune the value of λ. Based on our observation, this strategy could achieve the best performance for all the datasets.

There is another hyper-parameter K in the our model, which denotes the number of the nearest sample points in the feature space. We observe that most low-similarity pairwise samples have the similarity value close to zero and they have almost no influence to the final obtained S. In our implementation, we update S based on the nearest K pairwise samples. To prove this, we evaluate the performance with different $K = [0, 200]$ in Figure 5. It shows that the performance drops considerably when only a few (i.e., 0, 1, or 2) of the nearest neighbors are utilized for updating the adaptive graph. Meanwhile, $K > 200$ seems to have no distinctive negative influence to most

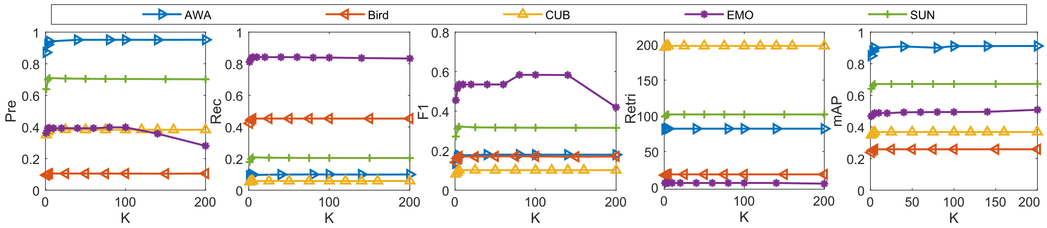
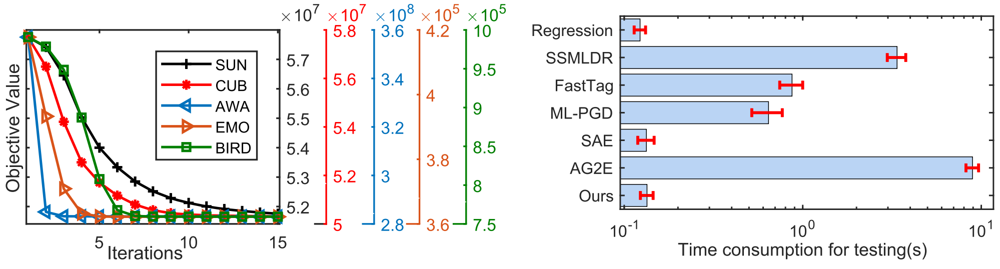


Fig. 5. Parameter sensitivity analysis of the graph nearest neighbor K in S optimization procedure. It shows that instead of optimizing S for all sample pairs, calculating several nearest sample pairs could achieve the similar performance. It indicates the robustness of adaptive graph and we can further reduce the computational complexity in training procedure by reducing the value of K.



(a) (b)

Fig. 6. (a) Objective function (i.e., Equation (11)) values as iteration increases. It illustrates the convergence of the optimization procedure. (b) Time consumption of all methods in the testing stage. It illustrates that our approach is one of the most efficient methods which is suitable for large-scale applications.

of the datasets. We observed that most of the elements in S are very close to 0, which means S is usually sparse after the optimization procedure. Thus, K does not have any negative influence on the final performance if K is great enough. From the result, we conclude that $K \geq 30$ is an appropriate value for most cases, and we do not need further parameter tuning for K, which reduces the unnecessary calculation on updating S without loss the performance for the final prediction.

4.8 Convergence Analysis

In the training stage, we utilize the ADMM [3] algorithm for solving the objective function. Specifically, the three target variables are alternatively optimized to its optimal point until the final objective loss is converged (i.e., Equation (11)). Considering multiple are optimized independently in the training stage, thus, it is difficult to theoretically guarantee the obtained solution is the global optimal point. In practice, we empirically analyzed the global convergence of our approach. The objective function value of Equation (11) is shown in Figure 6(a) as the ADMM iteration increases, and different colors denote all five datasets. From Figure 6(a), we observe that the objective function values significantly decrease in the first 10 iterations and become stable afterward. The result empirically indicates that our optimization strategy is effective and could converge in most real-world datasets.

4.9 Time Consumption

The time consumption of each method is illustrated in Figure 6(b). We can see from the results that our model associated with SAE [14] and regression approaches are the most efficient approaches.

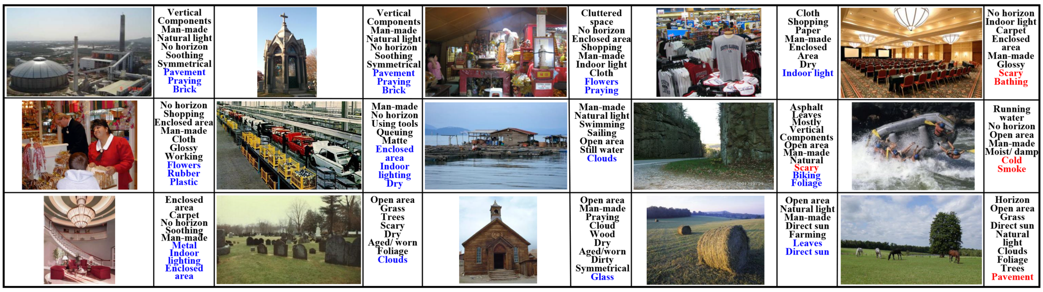


Fig. 7. Case study of the label prediction results from the SUN dataset. Black font means correct prediction and red font means incorrect prediction. In addition, blue font indicates the "correct" prediction based on our judgments while missing in ground truth.

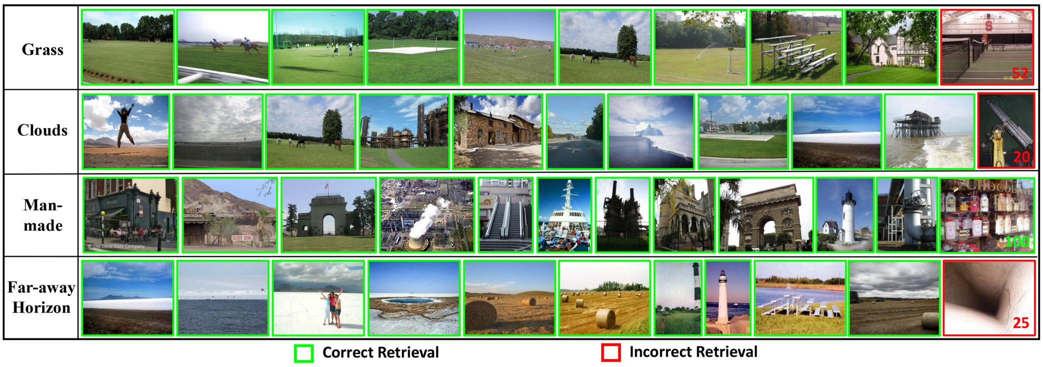


Fig. 8. Zero-shot image retrieval result from SUN dataset. Given a target retrieval label, the samples in the testing set which have the highest prediction score are selected. Green and red boxes are the correct and incorrect retrievals. The numbers in right bottom corner indicate the rankings of the samples.

The main explanation is that although in the training stage, our approach requires to alternatively optimize all the variables including P, S, and F. While, after the training procedure is finished, our approach could directly utilize the learned projection P to project new/unseen samples between visual and semantic/label spaces (Equation (2)). By this way, the inferring process could be degraded to a matrix multiplication operation without any extra computational costing calculations (e.g., eigen-decomposition). The complexity is $\mathbf{O}(n)$ where n is the input sample numbers. Our previous work, AG^{2}E [48], requires to update the entire adaptive graph based on labeled and unlabeled samples, which is both space and computational costly.

4.10 Image Annotation

Image annotation setting is evaluated in the SUN dataset. Figure 7 listed the sample images as well as the corresponding predicted labels. Different colors indicate different prediction results. Considering some samples have a large number of labels, we only list the top 15 labels for discussion. In Figure 7, the red font is the incorrect prediction and the black font is the correct prediction. Blue font indicates the "correct" prediction based on our judgments while missing in ground truth. Figure 7 illustrates that most of the prediction results are correct and our model is able to reveal several "missing" labels. It demonstrates the efficiency and effectiveness of our method.

4.11 Image Retrieval

Image retrieval setting is also evaluated. It retrieves specific images from a set of images [30]. In our implementation, the obtained P assigns labels to the candidate images. The candidate images are ranked based on the prediction confidence. Zero-shot setting is utilized which means the target image categories are unseen in the training stage. The retrieved samples are listed in Figure 8. Each row shows the retrieval label and the obtained images. The images with green and red boxes are the corrected and incorrect retrieval. We observe that our model effectively retrieves the target images even based on the target label even if the image categories are unseen in the training stage.

5 CONCLUSION

We designed a novel generic multi-label learning framework via AGMA in a semi-supervised learning scenario. It efficiently utilizes limited labeled samples associated with unlabeled samples to improve learning performance. In AGMA model, an adaptive similarity graph is learned to effectively obtain the intrinsic structure within the data; moreover, a marginalized strategy is explored to further augment the samples to reinforce the generalization and robustness of the learned model. An autoencoder is utilized to connect visual space and label space. Extensive experiments prove the usefulness of all designed modules in our framework, and demonstrate the high robustness, accuracy, and efficiency of our AMGA method.

REFERENCES

[1] Richard H. Bartels and G. W. Stewart. 1972. Solution of the matrix equation AX+XB=C [F4]. *ACM Communications* 15, 9 (1972), 820–826.

[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] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning* 3, 1 (2011), 1–122.

[4] Stephen Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press.

[5] Forrest Briggs, Xiaoli Z. Fern, Raviv Raich, and Qi Lou. 2013. Instance annotation for multi-instance multi-label learning. *ACM Transactions on Knowledge Discovery from Data* 7, 3 (Sept. 2013), 30 pages.

[6] F. Briggs, B. Lakshminarayanan, L. Neal, X. Z. Fern, R. Raich, S. J. K. Hadley, A. S. Hadley, and M. G. Betts. 2013. New methods for acoustic classification of multiple simultaneous bird species in a noisy environment. In *Proceeding of the IEEE International Workshop on Machine Learning for Signal Processing*. 1–8.

[7] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. 2006. *Semi-supervised Learning*. The MIT Press.

[8] Mingyu Chen and Alexander Hauptmann. 2007. Discriminative fields for modeling semantic concepts in video. In *Proceeding of the Large Scale Semantic Access to Content*. 151–166.

[9] Minmin Chen, Kilian Weinberger, Fei Sha, and Yoshua Bengio. 2014. Marginalized denoising auto-encoders for nonlinear representations. In *Proceeding of the International Conference on Machine Learning*. 1476–1484.

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

[11] Don Coppersmith and Shmuel Winograd. 1987. Matrix multiplication via arithmetic progressions. In *Proceeding of the ACM Symposium on Theory of Computing*. 1–6.

[12] Jia Deng, Wei Dong, Richard Socher, Lijia Li, Kai Li, and Feifei Li. 2009. ImageNet: A large-scale hierarchical image database. In *Proceeding of the IEEE Computer Vision and Pattern Recognition*. 248–255.

[13] Zhengming Ding and Yun Fu. 2014. Low-rank common subspace for multi-view learning. In *Proceeding of the IEEE International Conference on Data Mining*. 110–119.

[14] Tao Xiang Elyor Kodirov and Shagong Gong. 2017. Semantic autoencoder for zero-shot learning. In *Proceeding of the IEEE Computer Vision and Pattern Recognition*.

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

[16] Shantanu Godbole and Sunita Sarawagi. 2004. Discriminative methods for multi-labeled classification. In *Proceeding of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*. 22–30.

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

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

[19] Yumeng Guo, Fulai Chung, Guozheng Li, Jiancong Wang, and James C. Gee. 2019. Leveraging label-specific discriminant mapping features for multi-label learning. *ACM Transactions on Knowledge Discovery from Data* 13, 2 (Apr. 2019), 23 pages.

[20] Zhenyu Huang, Peng Hu, Joey Tianyi Zhou, Jiancheng Lv, and Xi Peng. 2020. Partially view-aligned clustering. *Proceeding of the Neural Information Processing Systems*.

[21] Shuiwang Ji, Lei Tang, Shipeng Yu, and Jieping Ye. 2010. A shared-subspace learning framework for multi-label classification. *ACM Transactions on Knowledge Discovery from Data* 4, 2 (May 2010), 29 pages.

[22] Z. Kang, H. Pan, S. C. H. Hoi, and Z. Xu. 2020. Robust graph learning from noisy data. *IEEE Transactions on Cybernetics* 50, 5 (2020), 1833–1843.

[23] Zhao Kang, Chong Peng, Ming Yang, and Qiang Cheng. 2017. Exploiting nonlinear relationships for top-N recommender systems. In *Proceeding of the IEEE International Conference on Big Knowledge*. 49–56.

[24] Zhao Kang, Liangjian Wen, Wenyu Chen, and Zenglin Xu. 2019. Low-rank kernel learning for graph-based clustering. *Knowledge-Based Systems* 163 (2019), 510–517. DOI: https://doi.org/10.1016/j.knosys.2018.09.009

[25] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. 2009. Learning to detect unseen object classes by between-class attribute transfer. In *Proceeding of the IEEE Computer Vision and Pattern Recognition*. 951–958.

[26] 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.

[27] Yingming Li, Ming Yang, Zenglin Xu, and Zhongfei Zhang. 2016. Learning with marginalized corrupted features and labels together. In *Proceeding of the AAAI Conference on Artificial Intelligence*. 1251–1257.

[28] Jing Liu, Mingjing Li, Weiying Ma, Qingshan Liu, and Hanqing Lu. 2006. An adaptive graph model for automatic image annotation. In *Proceeding of the ACM International Workshop on Multimedia Information Retrieval*. 61–70.

[29] Weiwei Liu, Ivor W. Tsang, and Klaus-Robert Müller. 2017. An easy-to-hard learning paradigm for multiple classes and multiple labels. *Journal of Machine Learning Research* 18, 94 (2017), 1–38.

[30] Weiwei Liu, Donna Xu, Ivor Tsang, and Wenjie Zhang. 2018. Metric learning for multi-output tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 51, 2 (2018), 408–422.

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

[32] Qianqian Ma and Alex Olshevsky. 2020. Adversarial crowdsourcing through robust rank-one matrix completion. In *Proceeding of the Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., 21841–21852.

[33] Laurens Maaten, Minmin Chen, Stephen Tyree, and Kilian Weinberger. 2013. Learning with marginalized corrupted features. In *Proceeding of the International Conference on Machine Learning*. 410–418.

[34] Laurens Maaten, Minmin Chen, Stephen Tyree, and Kilian Weinberger. 2014. Marginalizing corrupted features. *arXiv preprint arXiv:1402.7001* (2014).

[35] Feiping Nie, Guohao Cai, and Xuelong Li. 2017. Multi-view clustering and semi-supervised classification with adaptive neighbours. In *Proceeding of the AAAI Conference on Artificial Intelligence*. 2408–2414.

[36] Feiping Nie, Dong Xu, and Xuelong Li. 2012. Initialization independent clustering with actively self-training method. *IEEE Transactions on Systems, Man, and Cybernetics* 42, 1 (2012), 17–27.

[37] Feiping Nie, Sheng Yang, Rui Zhang, and Xuelong Li. 2019. A general framework for auto-weighted feature selection via global redundancy minimization. *IEEE Transactions on Image Processing* 28, 5 (2019), 2428–2438.

[38] Feiping Nie, Wei Zhu, and Xuelong Li. 2016. Unsupervised feature selection with structured graph optimization. In *Proceeding of the AAAI Conference on Artificial Intelligence*. 1302–1308.

[39] Mark Palatucci, Dean Pomerleau, Geoffrey E. Hinton, and Tom M. Mitchell. 2009. Zero-shot learning with semantic output codes. In *Proceeding of the Neural Information Processing Systems*. 1410–1418.

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

[41] Xi Peng, Hongyuan Zhu, Jiashi Feng, Chunhua Shen, Haixian Zhang, and Joey Tianyi Zhou. 2019. Deep clustering with sample-assignment invariance prior. *IEEE Transactions on Neural Networks and Learning Systems* 31, 11 (2019), 4857–4868.

[42] Guojun Qi, Xiansheng Hua, Yong Rui, Jinhui Tang, Tao Mei, and Hongjiang Zhang. 2007. Correlative multi-label video annotation. In *Proceedings of the 15th ACM International Conference on Multimedia*. 17–26.

[43] Martha Roseberry, Bartosz Krawczyk, and Alberto Cano. 2019. Multi-label punitive KNN with self-adjusting memory for drifting data streams. *ACM Transactions on Knowledge Discovery from Data* 13, 6 (Nov. 2019), 31 pages.

[44] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations.*

[45] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceeding of the IEEE Computer Vision and Pattern Recognition.* 1–9.

[46] Konstantinos Trohidis, Grigorios Tsoumakas, George Kalliris, and Ioannis P. Vlahavas. 2008. Multi-label classification of music into emotions. In *Proceeding of the ISMIR.* 325–330.

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

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

[49] Lichen Wang, Zhengming Ding, and Yun Fu. 2018. Learning transferable subspace for human motion segmentation. In *Proceeding of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[50] Lichen Wang, Zhengming Ding, and Yun Fu. 2018. Low-rank transfer human motion segmentation. *IEEE Transactions on Image Processing* 28, 2 (2018), 1023–1034.

[51] Lichen Wang, Zhengming Ding, Zhiqiang Tao, Yunyu Liu, and Yun Fu. 2019. Generative multi-view human action recognition. In *Proceeding of the IEEE International Conference on Computer Vision.* 6212–6221.

[52] Wei Wang, Yan Yan, Feiping Nie, Shuicheng Yan, and Nicu Sebe. 2018. Flexible manifold learning with optimal graph for image and video representation. *IEEE Transactions on Image Processing* 27, 6 (2018), 2664–2675.

[53] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. 2010. Caltech-UCSD Birds 200. Technical Report. California Institute of Technology. CNS-TR-2010-001. 2010.

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

[55] Hao Yang, Joey Tianyi Zhou, and Jianfei Cai. 2016. Improving multi-label learning with missing labels by structured semantic correlations. In *Proceeding of the European Conference on Computer Vision.* Springer, 835–851.

[56] Zhengjun Zha, Tao Mei, Jingdong Wang, Zengfu Wang, and Xiansheng Hua. 2009. Graph-based semi-supervised learning with multiple labels. *Journal of Visual Communication and Image Representation* 20, 2 (2009), 97–103.

[57] Yu Zhang and Dityan Yeung. 2013. Multilabel relationship learning. *ACM Transactions on Knowledge Discovery from Data* 7, 2 (Aug. 2013), 30 pages.

[58] Yin Zhang and Zhihua Zhou. 2010. Multilabel dimensionality reduction via dependence maximization. *ACM Transactions on Knowledge Discovery from Data* 4, 3 (2010), 21 pages.

[59] Joey Tianyi Zhou, Ivor W. Tsang, Sinno Jialin Pan, and Mingkui Tan. 2019. Multi-class heterogeneous domain adaptation. *Journal of Machine Learning Research* 20, 57 (2019), 1–31.

[60] Joey Tianyi Zhou, Heng Zhao, Xi Peng, Meng Fang, Zheng Qin, and Rick Siow Mong Goh. 2018. Transfer hashing: From shallow to deep. *IEEE Transactions on Neural Networks and Learning Systems* 29, 12 (2018), 6191–6201.

[61] Tianyi Zhou, Shengjie Wang, and Jeff Bilmes. 2020. Time-consistent self-supervision for semi-supervised learning. In *Proceeding of the International Conference on Machine Learning.* 11523–11533.

[62] Xiaojin Zhu. 2005. Semi-supervised learning literature survey. *Technical Report 1530, University of Wisconsin-Madison.*

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

Received October 2020; revised February 2021; accepted February 2021