A Transferable Generative Framework for Multi-Label Zero-Shot Learning

Peirong Ma, Zhiquan He, Wu Ran, *Graduate Student Member, IEEE*, and Hong Lu, *Member, IEEE*

***Abstract*— Multi-label zero-shot learning (MLZSL) is a more realistic and challenging task than single-label zero-shot learning (SLZSL), which aims to recognize multiple unseen classes in a single image. To adapt generative models to the MLZSL task and better recognize multiple unseen object categories in an image, this paper proposes a Transferable Generative Framework (TGF), which consists of a Multi-Label Semantic Embedding Autoencoders (SEAs), a Semantic-Related Multi-Label Feature Transformation Network (FTN) and a Multi-Label Feature Generation Networks (FGNs). First, SEAs adaptively encodes the class-level word vectors corresponding to each sample containing different number of classes into sample-level semantic embeddings with the same dimension. Then, FTN transforms global features extracted by a CNN pre-trained on single-label images into features that are semantic-related and more suitable for multi-label classification. Finally, FGNs generates both global and local features to better recognize the dominant and minor object categories in a multi-label image, respectively. Extensive experiments on three benchmark datasets show that TGF significantly outperforms state-of-the-arts. Specifically, compared with the previous best generative MLZSL method (*i.e.*, Gen-MLZSL), TGF improves the mAP of the ZSL (GZSL) task by 5.4% (6.9%), 20.5% (27.9%), and 2.4% (3.9%) on NUS-WIDE, Open Images, and MS-COCO datasets, respectively.**

***Index Terms*— Multi-label zero-shot learning, multi-label semantic embedding autoncoders, multi-label feature transformation networks, multi-label feature generation networks.**

(a) Single-label images form AWA2

(b) Multi-label images from NUS-WIDE

Fig. 1. Some examples from the SLZSL dataset AWA2 [13] and the MLZSL dataset NUS-WIDE [14]. In AWA2, an image contains only one category (label), while in NUS-WIDE, an image contains a variable numbers of multiple categories (dominant and small object categories).

I. Introduction

THE goal of the traditional multi-label classification task [1], [2], [3], [4], [5], [6] is to recognize all seen classes in an image, that is, the class labels of the training and test set are the same in this task. However, in the real world, a scene may contain both seen and unseen categories, which greatly limits the practical application of traditional multi-label classification algorithms. To address this problem, multi-label zero-shot learning (MLZSL) [7], [8], [9], [10], [11], [12] has attracted a lot of interest in recent years. MLZSL aims to recognize multiple unseen classes in an image at test time; while multi-label generalized zero-shot learning (MLGZSL) is a more realistic and challenging variant of MLZSL, where test images can belong to multiple seen and unseen classes, not just unseen classes. In this paper, we address both the MLZSL and MLGZSL tasks.

Compared with MLZSL, single-label zero-shot learning (SLZSL) has made remarkable progress [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28]. SLZSL transfers knowledge from seen classes to unseen classes through semantic information (such as attributes [29], [30], word vectors [31], [32], or sentence descriptions [33]) shared among all classes to achieve unseen class recognition. Early SLZSL methods [33], [34], [35], [36], [37] construct an embedding model to learn the cross-modal mapping between visual feature space and semantic space. However, since the seen classes and the unseen classes are disjoint, the embedding model learned only on the seen classes will produce a large bias when directly used for unseen classes prediction (*i.e.*, projection domain shifts [38]). As a result, embedding-based SLZSL methods usually perform poorly. In recent years,

Manuscript received 22 March 2023; revised 14 July 2023 and 6 September 2023; accepted 8 October 2023. Date of publication 16 October 2023; date of current version 9 May 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62072112, in part by the National Key Research and Development Program of China under Grant 2020AAA0108301, and in part by the Key Area Support Plan of Guangdong Province for Jihua Laboratory under Grant X190051TB190. This article was recommended by Associate Editor M. Shojafar. *(Corresponding author: Hong Lu.)*

The authors are with the Shanghai Key Laboratory of Intelligent Information Processing, School of Computer Science, Fudan University, Shanghai 200438, China (e-mail: prma20@fudan.edu.cn; zqhe22@m.fudan.edu.cn; wran21@m.fudan.edu.cn; honglu@fudan.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2023.3324648.

Digital Object Identifier 10.1109/TCSVT.2023.3324648

generative models such as generative adversarial network (GAN) [39], [40] and variational autoencoder (VAE) [41] have been widely used in SLZSL and become the mainstream methods [15], [16], [17], [18], [19], [20], [21]. The generative methods treat SLZSL as an unseen class samples missing problem and train a generative model to synthesize unseen image features under the condition of the class-specific semantic embedding. Then, with the real seen class features and the synthetic unseen class features, SLZSL is transformed into a standard supervised classification task.

Generative SLZSL methods address the data imbalance between seen and unseen classes and achieve better performance than embedding-based SLZSL methods. In addition, the process of synthesizing unseen class samples conditioned on the corresponding semantic embedding by generative methods is more similar to the process of human learning to recognize a novel unseen class. For example, a person who has never seen a zebra before can picture it in his mind based on some semantic descriptions of zebras (*e.g.*, black and white, striped and horse-like in appearance), and when he sees a real zebra, he can recognize it accurately. This further illustrates that generative methods are more advanced methods than embedding-based methods. However, almost all existing MLZSL methods are non-generative [7], [8], [10], [11]. They only use the image features and semantic embeddings (*e.g.*, word vectors) of the seen classes to train a classification model and directly use it to recognize unseen classes. This suffers from the same drawback as the embedding-based SLZSL method. The important reasons that limit the application of the generative SLZSL models in MLZSL are that: **Firstly**, an image contains only one class (label) in SLZSL (as shown in Figure 1a), while in MLZSL, an image contains a variable number of multiple classes (as shown in Figure 1b), and a variable number of classes implies a variable number of class semantic embeddings, which makes it impossible to determine the input layer size of the generative model. **Secondly**, in SLZSL, we generate single-label features according to the class-specific semantic embedding; while in MLZSL, we need a semantic embedding containing multi-label information to generate multi-label features.

To solve the problems mentioned above and better adapt generative model to MLZSL task, we propose a novel Transferable Generative Framework (TGF) consisting of a Multi-Label Semantic Embedding Autoencoders (SEAs), a Semantic-Related Multi-Label Feature Transformation Network (FTN) and a Multi-Label Feature Generation Networks (FGNs). Figure 2a presents an overview of the proposed TGF. Specifically, SEAs adaptively encodes the class-level word vectors of each sample containing different number of classes into sample-level multi-label semantic embeddings with the same dimension. Using these sample-level semantic embeddings, we can transfer any advanced generative SLZSL model to solve MLZSL task. In addition, FTN transforms global features extracted from a CNN pre-trained on single-label images into an embedding space that is semantic-related and more suitable for multi-label classification. Finally, we train our FGNs (composed of FGN-t and FGN-l) using the multi-label semantic embeddings obtained via SEAs, semantic-related multi-label features obtained via FTN, and the original local features. The feature generation process is shown in Figure 2b. After training, given a multi-label combination, we first obtain a multi-label semantic embedding using SEAs, and then use this multi-label semantic embedding as the conditional input of FGN-t and FGN-l to synthesize global and local features to exploit their ability to recognize the dominant and small object categories in a multi-label image, respectively.

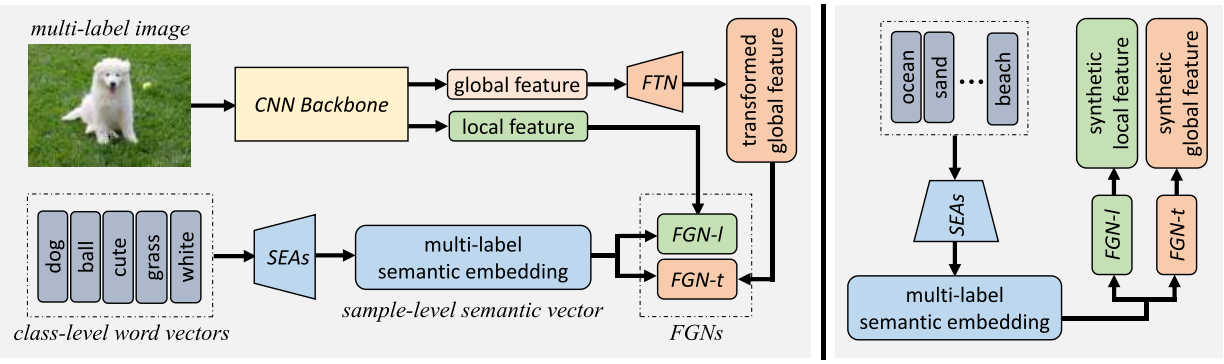
The contributions are as follows:

- A novel Transferable Generative Framework (TGF) is proposed for MLZSL, which combines the advantages of SEAs, FTN and FGNs.
- SEAs can encode class-level word vectors into sample-level multi-label semantic embedding to break the limitation of generative models applied to MLZSL, and SEAs allows us to transfer any existing generative SLZSL model to solve MLZSL task.
- FTN transforms the global features extracted from a pre-trained CNN into features that semantic-related and more suitable for multi-label classification. Alternatively, other advanced traditional multi-label classification network can be employed as our FTN (an MLP with only one hidden layer in this paper) to further improve the performance.
- FGNs is the first model to generate both global and local features for MLZSL, which helps us better recognize dominant classes and small classes in a multi-label image, respectively.
- Extensive experiments are conducted on three MLZSL benchmark datasets, and the results show that the proposed method significantly outperforms the state-of-the-art. In detail, compared with Gen-MLZSL [9], the previous best generative MLZSL method, our TGF achieves absolute gains of 5.4%, 20.5%, and 2.4% in terms of mAP for the ZSL task on NUS-WIDE, Open Images, and MS-COCO datasets, respectively. TGF also achieves consistent improvements under the more challenging GZSL setting, obtaining absolute gains of 6.9%, 27.9%, and 3.9% in mAP on these datasets, respectively.

The rest of this paper is organized as follows. We first review the related work in Section II. Then, Section III presents the proposed method. After that, the experiments are reported in Section IV. Finally, Section V concludes this paper.

II. Related Work

In recent years, both multi-label classification [3], [4], [5], [6], [42] and zero-shot learning [15], [17], [21], [33], [35], [36] have attracted huge research attention and made significant progress. However, there are relatively few studies on more realistic MLZSL, which is still a very challenging task. Among the existing MLZSL methods, Fast0Tag [43] trains a network to estimate a single principal direction for each image so that the word vectors of relevant labels rank ahead of irrelevant labels. Inspired by the way humans exploit semantic knowledge among objects of interest, SKG [7] proposes a knowledge graph-based framework to describe



(a) Overview of the proposed TGF (b) Feature generation with trained TGF

Fig. 2. The proposed TGF contains a Multi-Label Semantic Embedding Autoencoders (SEAs), a Semantic-Related Multi-Label Feature Transformation Network (FTN), and a Multi-Label Feature Generation Networks (FGNs), where the FGNs consists of a global feature generation network (FGN-t) and a local feature generation network (FGN-l).

the relationships between multiple labels. LESA [8] proposes a MLZSL framework based on a shared multi-attention mechanism, which can recognize multiple unseen labels in an image and find the relevant regions of each label. BiAM [10] utilizes a bi-level attention module to obtain discriminative features by combining region and scene context information, and then uses a compatibility function to perform region-based classification. SDL [11] argues that using a single embedding vector to represent an image is insufficient to accurately rank the relevant seen and unseen labels. It properly exploits and handles the semantic diversity of labels in each image by allowing multiple principal directions.

However, like the earlier SLZSL methods, the above-mentioned MLZSL methods are all non-generative, and they deal with unseen classes in an indirect way, *i.e.*, they use the classification model learned only on the seen classes directly for the unseen classes recognition, which inevitably suffers from projection domain shifts. Moreover, as mentioned in Section I, generative methods are more advanced methods, and the current state-of-the-art SLZSL methods are almost all generative in nature. Generative methods transform SLZSL into a traditional classification task by synthesizing unseen class samples, and dealing with unseen classes in a direct manner. The difference in the number of classes (labels) contained in each sample in a multi-label classification dataset is the main reason that limits the application of generative models to MLZSL. GCDN [44] and MUCO [45] conduct zero-shot classification experiments on three widely used SLZSL datasets (*i.e.*, SUN [46], CUB [47], and AWA1 [48]), and they use the attribute vector of the corresponding category to synthesize single-label features of this category. As in SLZSL, Gen-ADA [12] insists on considering only one class semantic embedding at a time as the conditional input of the generator to avoid this limitation, so it also generates all single-label features without considering the correlation between multiple labels in an image, which is not conducive to multi-label classification. On the other hand, in order to make the network obtain a uniform input and output size, Gen-MLZSL [9] averages the word vectors corresponding to each sample at the input of the generator (*i.e.*, ALF) and averages the single-label features generated separately by the word vectors corresponding to each sample at the output of the generator (*i.e.*, FLF). However, simply averaging the input and output destroys the original data structure and distribution (especially for the sample containing a large number of categories) and inevitably loses much discriminative information that is beneficial for classification. Moreover, after ALF and FLF, Gen-MLZSL also requires a complex attention module for feature fusion, which is time-consuming and memory-consuming.

To fundamentally break the limitations of generative models applied to MLZSL and generate discriminative multi-label features to better recognize multiple unseen classes in an image, we construct a novel TGF consisting of three components: SEAs, FTN and FGNs. SEAs can generate a sample-level semantic embedding containing multi-label information according to the class-level word vectors corresponding to each sample for subsequent training of FTN and FGNs. With our SEAs, any existing generative SLZSL model can be directly transferred to solve the MLZSL task. In addition, previous works [7], [8], [9], [10], [12], [43] on MLZSL all use visual features extracted from a CNN (VGG [49] or ResNet [50]) pre-trained on single-label images (ImageNet 1K [51]) for multi-label classification, and the correlation between these features and the corresponding class semantic embeddings (GloVe [32] vector) is weak. Therefore, we train the FTN to obtain semantic-related and more suitable visual features for multi-label classification. Finally, we train the FGNs to generate both global and local features to better recognize dominant and minor classes in multi-label images, respectively.

III. Methodology

This section first presents the problem definition, then gives an overview of the proposed TGF, and finally details the three components (*i.e.*, SEAs, FTN and FGNs) in the TGF respectively.

A. Problem Definition

Let the entire label set: $\mathcal{C} = \mathcal{C}^S \cup \mathcal{C}^U$, where $\mathcal{C}^S$ denotes the seen label set with training annotations, $\mathcal{C}^U$ denotes the

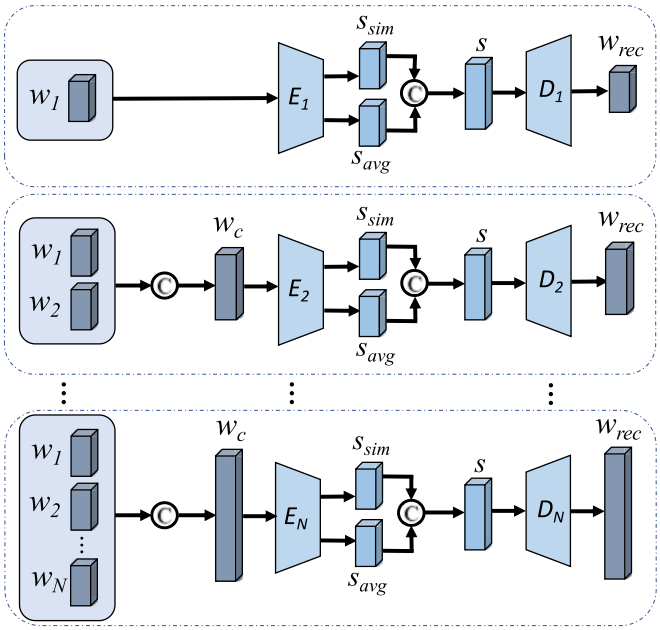


Fig. 3. Training for the proposed Multi-Label Semantic Embedding Autoencoders (SEAs). w_i denotes the i-th word vector corresponding to a multi-label image, and N denotes the maximum number of word vectors (classes) present in a single image in the multi-label dataset. © denotes the concatenating operation.

unseen label set without training images, and $\mathcal{C}^S \cap \mathcal{C}^U = \phi$. Let $\{(x_m, y_m); m = 1, 2, \ldots, M\}$ denote M multi-label training images, where x_m denotes the m-th image and $y_m \subseteq \mathcal{C}^S$ denotes the set of labels present in this image. Each label (class) corresponds to a semantic word vector $\{w_y\}_{y \in \mathcal{C}}$. The goal of MLZSL is to assign relevant unseen labels $y_i \subseteq C^U$ to a given test image x_i, while the goal of the more realistic and challenging MLGZSL is to assign relevant seen and unseen labels $y_i \subseteq C$ for a given test image x_i.

B. The Overall Framework

As shown in Figure 2a, the proposed TGF has three components: a Multi-Label Semantic Embedding Autoencoders (SEAs), a Semantic-Related Multi-Label Feature Transformation Network (FTN), and a Multi-Label Feature Generation Networks (FGNs). In addition, the FGNs consists of a global feature generation network (FGN-t) and a local feature generation network (FGN-l). The training of our TGF is divided into three stages. First, we train the SEAs, as shown in Figure 3. Then, we train the FTNs with frozen SEAs, as shown in Figure 4. Finally, we train the FGNs with frozen SEAs and FTNs, as shown in Figure 5. More specifically, we first train SEAs to adaptively encode the corresponding class-level word vectors for each sample containing different number of classes into sample-level multi-label semantic embeddings with the same dimension. Then, we train our FTN with these multi-label semantic embeddings and the widely used multi-label classification loss (*i.e.*, Asymmetric Loss [3]) to transform the global features extracted from a CNN backbone pre-trained on single-label images into semantic-related and more suitable features for multi-label classification. Finally, with the multi-label semantic embeddings obtained by SEAs, the semantic-related multi-label features obtained by FTN, and the original local features, we train our FGNs that can generate both global and local features. The feature generation process is shown in Figure 2b. After training, given a multi-label combination, we first obtain a multi-label semantic embedding with SEAs, and then use this multi-label semantic embedding as the conditional input of FGN-t and FGN-l to synthesize global and local features to train the final MLZSL classifiers. The following will introduce the proposed TGF in detail.

C. Multi-Label Semantic Embedding Autoencoders

In a multi-label dataset, it is assumed that the number of categories that may be contained in an image is denoted as $n \in \{1, 2, \ldots, N\}$. The variability of n leads to changes in the number of word vectors corresponding to an image, which makes it impossible for us to construct a generative model with a fixed input size as in SLZSL to generate single-label image features conditioned on the word vector of a single category. What we need is a semantic embedding containing multi-label information, which is then used as a condition to generate multi-label features. To this end, we construct a Multi-Label Semantic Embedding Autoencoders (SEAs), which train a SEA for each training sample set containing the same class number n, respectively. As shown in Figure 3, each SEA consists of an encoder E_n and a decoder D_n, which can encode a set of class-level word vectors corresponding to a multi-label image into a sample-level semantic embedding that represents the semantic description of all the positive labels in this image. Specifically, E_n takes w_c as input, and encodes it as s_{sim} and s_{avg}:

$$s_{sim},\ s_{avg} = E_n\left(w_c\right), \tag{1}$$

where $\{w_1, w_2, \ldots, w_n\}$ denotes the set of class-level word vectors corresponding to a multi-label image, and $w_c = concat(w_1, w_2, \ldots, w_n)$. s_{sim} and s_{avg} are the latent vectors output by E_n, and we use the concatenation of s_{sim} and s_{avg}, *i.e.*, $s = cat(s_{sim}, s_{avg})$, as the final multi-label semantic embedding.

To preserve class-specific semantic information, we maximize the cosine similarity between s_{sim} and $w_i \in \{w_1, w_2, \ldots, w_n\}$ respectively:

$$\mathcal{L}_{sim} = \mathbb{E}[\frac{1}{n}\sum_{i=1}^{n} \|1 - \cos\left(s_{sim}, w_i\right)\|^2]. \tag{2}$$

In addition, to learn the correlation between classes in a multi-label image, we use a mean square error (MSE) loss to optimize the following objective:

$$\mathcal{L}_{avg} = \mathbb{E}[\left\|s_{avg} - \bar{w}\right\|^2], \qquad \bar{w} = \frac{1}{n}\sum_{i=1}^{n} w_i. \tag{3}$$

Meanwhile, D_n reconstructs $s = concat(s_{sim}, s_{avg})$ into w_c to prevent information loss. Here we also use MSE loss to formulate the reconstruction objective:

$$\mathcal{L}_{rec} = \mathbb{E}[\|w_{rec} - w_c\|^2], \qquad w_{rec} = D_n(s). \tag{4}$$

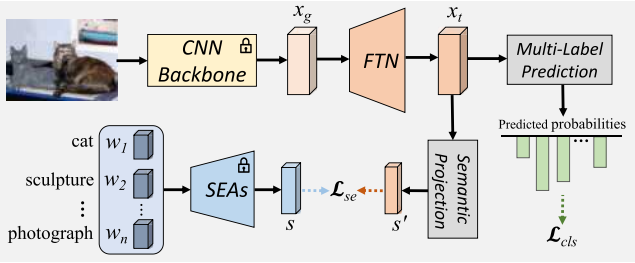


Fig. 4. Training for the Proposed Semantic-Related Multi-Label Feature Transformation Network (FTN). The lock symbol indicates a frozen network.

The overall loss of the proposed SEA is as follows:

$$\mathcal{L}_{SEA} = \mathcal{L}_{sim} + \mathcal{L}_{avg} + 0.1 * \mathcal{L}_{rec}. \tag{5}$$

We found that $N = 20$ for NUS-WIDE, $N = 15$ for Open Images and $N = 8$ for MS COCO already cover 99% of the training set samples. Therefore, in the experiments, we set $N = 20$ for NUS-WIDE, $N = 15$ for Open Images and $N = 8$ for MS COCO, *i.e.*, we train 20 SEA for NUS-WIDE, 15 SEA for Open Images and 8 SEA for MS COCO, respectively. This is efficient (*e.g.*, it takes about one minute to train 20 SEAs for 10 epochs on NUS-WIDE with one NVIDIA GeForce RTX 3090 GPU) because our SEA uses a very simple network architecture (an MLP with only one hidden layer). After training, SEAs can generate discriminative multi-label semantic embeddings s according to the set of word vectors corresponding to any multi-label combination. With these discriminative s, we can transfer any existing generative SLZSL model to solve the MLZSL task.

D. Semantic-Related Multi-Label Feature Transformation Network

The previous MLZSL methods [7], [8], [9], [10], [12], [43] all use a CNN (VGG [49] or ResNet [50]) pre-trained on the single-label image dataset ImageNet 1K [51] to extract multi-label features. We argue that the features extracted from these networks trained for single-label classification are not suitable for multi-label classification. Furthermore, previous works use the GloVe model [32] trained on Wikipedia articles to extract semantic word vectors, which are poorly correlated with the corresponding visual features. As a result, the generalization is poor when conditioned on these semantic word vectors to synthesize unseen samples using the generator trained on seen classes.

To address these issues, we train a flexible and efficient Semantic-Related Multi-Label Feature Transformation Network (FTN) on seen classes to obtain visual features that are semantic-related and more suitable for multi-label classification. As shown in Figure 4, FTN maps the global image features x_g extracted by the pre-trained CNN to a new embedding space and obtains the transformed features x_t. Then, x_t is fed into a multi-label prediction layer (*i.e.*, a multi-label classifier) and outputs the predicted label probabilities $\{p_1, p_2, \ldots, p_K\}$, where K represents the total number of seen categories. Here, we use the Asymmetric Loss (ASL) [3] commonly used in multi-label classification to optimize:

$$\mathcal{L}_{cls} = -\frac{1}{M}\sum_{m=1}^{M}\sum_{k=1}^{K}\left[y_{m,k}L_+ + \left(1 - y_{m,k}\right)L_-\right], \tag{6}$$

with

$$\begin{cases} L_+ = \left(1 - p_{m,k}\right)^{\gamma_+}\log\left(p_{m,k}\right) \\ L_- = \left(p_t\right)^{\gamma_-}\log\left(1 - p_t\right), \end{cases} \tag{7}$$

$$p_t = \max\left(p_{m,k} - t, 0\right), \tag{8}$$

where $y_{m,k}$ denotes the one-hot label for class k of sample m, and L_+ and L_- denote the positive and negative loss parts, respectively. t is a hyperparameter, and when the probability of negative is less than t, it will be set to 0 directly. In this paper: $\gamma_- = 4$, $\gamma_+ = 0$ and $t = 0.05$ for NUS-WIDE and MS COCO; $\gamma_- = 4$, $\gamma_+ = 3$ and $t = 0.1$ for Open Images.

In addition, to enhance the consistency between the transformed visual features x_t and the corresponding multi-label semantic embeddings s, we also map x_t to s through a semantic projection layer. In this process, optimize:

$$\mathcal{L}_{se} = \mathbb{E}[\left\|s - s'\right\|^2]. \tag{9}$$

The overall loss of the proposed FTN is as follows:

$$\mathcal{L}_{FTN} = \mathcal{L}_{cls} + \mathcal{L}_{se}. \tag{10}$$

For simplicity, we use an MLP with only one hidden layer as our FTN, which already achieves state-of-the-art performance. This proves that the performance gains come from our novel ideas, not complex network architectures. But it is worth mentioning that the proposed FTN can be replaced by other more complex and advanced standard multi-label classification networks to further improve the performance, which we verify in section IV-H.3.

E. Multi-Label Feature Generation Networks

With the multi-label semantic embeddings s obtained from SEAs and the semantic-related multi-label features x_t obtained from FTN, we can employ any existing generative SLZSL model as our Multi-Label Feature Generation Networks (FGNs). For a simple and fair comparison, we use f-VAEGAN [15] as our generative backbone, as in Gen-MLZSL [9]. As shown in Figure 5, FGNs contain a global feature generation network (FGN-t) and a local feature generation network (FGN-l), where the FGN-t consists of a conditional VAE (C-VAE) and a conditional Wasserstein-GAN (C-WGAN), it is achieved by sharing the generator of C-WGAN and the decoder of C-VAE. Specifically, C-VAE includes an encoder E_t and a decoder/generator G_t. E_t encodes the input feature x_t and condition s into latent variable z_t, while G_t reconstructs the input x_t from the latent vector z_t and condition s. In this process, optimize:

$$\mathcal{L}^t_{VAE} = \mathcal{L}^t_{KL} + \mathcal{L}^t_{REC}, \tag{11}$$

with

$$\mathcal{L}^t_{KL} = D_{KL}(q(z_t|x_t, s)||p(z_t|s)), \tag{12}$$

$$\mathcal{L}^t_{REC} = -\mathbb{E}_{q(z_t|x_t,s)}\left[\log p(x_t|z_t, s)\right], \tag{13}$$

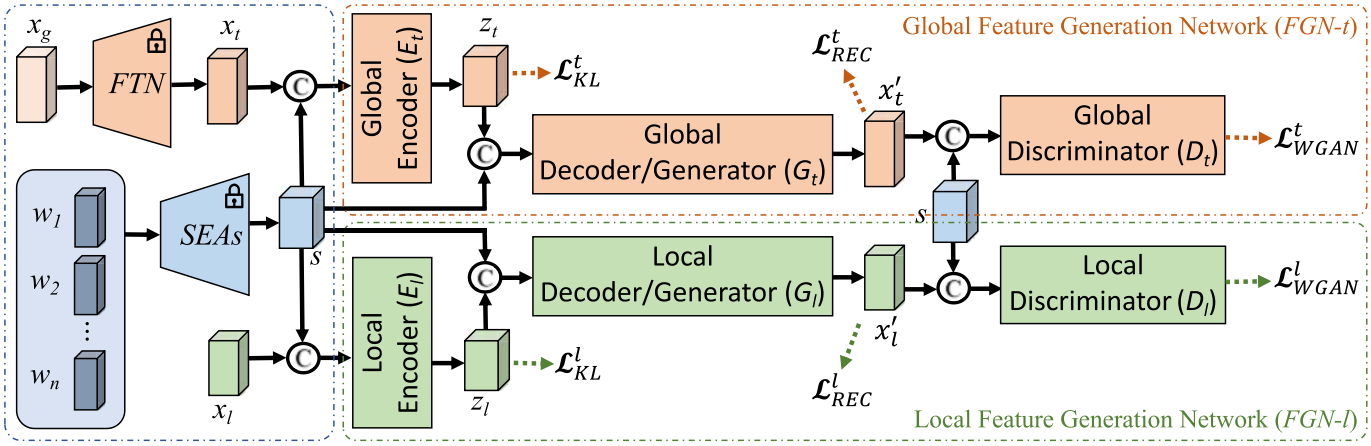


Fig. 5. Training for the Proposed Multi-Label Feature Generation Networks (FGNs). © denotes the concatenating operation. The lock symbol indicates a frozen network.

where D_{KL} is the Kullback-Leibler (KL) divergence, $p(z_t|s)$ is a prior distribution, which is assumed to be $\mathcal{N}(0, 1)$, and $\mathcal{L}^t_{REC}$ is the reconstruction loss.

C-WGAN consists of a generator/decoder G_t and a discriminator D_t. G_t utilizes random noise z and corresponding multi-label semantic embedding s to synthesize multi-label feature x_t. D_t takes as input a pair of input feature x_t and semantic embedding s, and outputs a real value representing the degree of realness or fakeness of the input feature. C-WGAN is learned by optimizing the following objectives:

$$\mathcal{L}^t_{WGAN} = \mathbb{E}\left[D(x'_t, s)\right] - \mathbb{E}\left[D\left(x_t, s\right)\right] + \gamma\mathbb{E}[(\|\nabla_{\tilde{x}_t} D(\tilde{x}_t, s)\|_2 - 1)^2], \tag{14}$$

where $x'_t = G_t(z, s)$ is the synthetic feature, $\tilde{x}_t = \eta x_t + (1 - \eta)x'_t$ with $\eta \sim U(0, 1)$, and $\gamma = 10$ is the penalty coefficient.

Additionally, as shown in Figure 1, unlike a single-label image that contains only one dominant class that occupies a large image area, a multi-label image may also contain several small classes. Reference [52] visualizes the feature maps output by each layer of CNN and concludes that the shallow features of CNN contain more image details (such as color, edge, contour, and texture), while the deep features are more discriminative and generalizable. In addition, [53] addresses the scale variations problem in crowd counting by extracting hierarchical features at different scales. Inspired by these work, we argue that shallow features containing more details are more beneficial to recognize small object categories in a multi-label image (**region-level local information**), while deeper features focus on dominant object categories that occupy larger areas of the image (**image-level global information**). Therefore, in addition to the global features x_g (which are transformed into x_t via FTN) extracted from the 2048-dim top pooling units of ResNet101 [50] pre-trained on ImageNet 1K [51], we also extract region-based features (of size h, w = 14 and d_r = 1024) from the layer conv4_23 of ResNet101 and directly average pool them as local features x_l with d_l=1024. We construct FGN-l for local features with the same architecture as FGN-t for global features, as depicted in Figure 5. Therefore, the overall loss of our FGNs is:

$$\mathcal{L}_{FGNs} = (\mathcal{L}^t_{VAE} + \alpha\mathcal{L}^t_{WGAN}) + \beta(\mathcal{L}^l_{VAE} + \alpha\mathcal{L}^l_{WGAN}), \tag{15}$$

where α is the hyperparameter that controls the loss weights of C-VAE and C-WGAN, and β is the hyperparameter that controls the loss weights of FGN-t and FGN-l. We set α = 10 and β = 1 for both NUS-WIDE and Open Images.

After training on the seen classes, our FGNs is able to synthesize discriminative global features x_t and local features x_l for the unseen classes. Subsequently, x_t and x_l are used for the training of global and local classifiers, respectively. The final prediction is determined by the sum of the predicted label scores of the two classifiers.

IV. Experiments

In this section, we first introduce the experimental setup and then compare the performance of our TGF with other state-of-the-art works in this field. Next, we make a detailed and comprehensive comparison (including: mAP improvement comparison, qualitative results and t-SNE visualization) between TGF and the previous state-of-the-art generative method Gen-MLZSL [9]. Subsequently, we compare the results of TGF using different CNN backbones and class embeddings. After that, we perform ablation studies and hyper-parameters sensitivity analysis for the components in TGF and some major objective functions. Finally, we evaluate our TGF on a novel Open-vocabulary multi-label classification (OVML) task.

A. Experimental Setup

1) Datasets: We evaluate the proposed TGF on three multi-label zero-shot benchmark datasets: NUS-WIDE [14], Open Images [57], and MS COCO [58]. The **NUS-WIDE** dataset includes 269,648 images, each with 81 human-annotated labels and 925 labels extracted from Flicker user tags. Consistent with previous work [8], [9], [10], [11], 925 and 81 labels are used as seen and unseen classes, respectively. The**Open Images (v4)** dataset consists of nearly 9 million training

TABLE I

STATE-OF-THE-ART COMPARISON FOR MLZSL AND MLGZSL TASKS ON THE NUS-WIDE AND OPEN IMAGES DATASETS. TOP: NON-GENERATIVE METHODS; BOTTOM: GENERATIVE METHODS. WE REPORT THE RESULTS IN TERMS OF MAP, AS WELL AS PRECISION (P), RECALL (R), AND F1 SCORE AT $K \in \{3, 5\}$ FOR NUS-WIDE AND $K \in \{10, 20\}$ FOR OPEN IMAGES. THE BEST AND THE SECOND BEST RESULTS ARE MARKED IN BOLD AND RED, RESPECTIVELY. SINCE BIAM [10] REPORTS WEIGHTED-MAP ON OPEN IMAGES, FOR A FAIR COMPARISON WITH OTHER METHODS, WE RE-IMPLEMENTED BIAM'S MAP ON OPEN IMAGES USING THE AUTHOR'S OFFICIAL CODE, THE RESULTS ARE HIGHLIGHTED WITH SYMBOL *. MEANWHILE, OUR METHOD CAN ACHIEVE HIGHER WEIGHTED-MAP THAN BIAM (86.4 FOR ZSL AND 88.5 FOR GZSL)

	Methods	Task	NUS-WIDE (#seen / #unseen = 925 / 81)							Open Images (#seen / #unseen = 7186 / 400)						
			K=3			K=5			mAP	K=10			K=20			mAP
			P	R	F1	P	R	F1		P	R	F1	P	R	F1	
NON-GEN	CONSE [54]	ZSL	17.5	28.0	21.6	13.9	37.0	20.2	9.4	0.2	7.3	0.4	0.2	11.3	0.3	40.4
		GZSL	11.5	5.1	7.0	9.6	7.1	8.1	2.1	2.4	2.8	2.6	1.7	3.9	2.4	43.5
	LabelEM [55]	ZSL	15.6	25.0	19.2	13.4	35.7	19.5	7.1	0.2	8.7	0.5	0.2	15.8	0.4	40.5
		GZSL	15.5	6.8	9.5	13.4	9.8	11.3	2.2	4.8	5.6	5.2	3.7	8.5	5.1	45.2
	Fast0Tag [43]	ZSL	22.6	36.2	27.8	18.2	48.4	26.4	15.1	0.3	12.6	0.7	0.3	21.3	0.6	41.2
		GZSL	18.8	8.3	11.5	15.9	11.7	13.5	3.7	14.8	17.3	16.0	9.3	21.5	12.9	45.2
	One Attention per Label [56]	ZSL	20.9	33.5	25.8	16.2	43.2	23.6	10.4	-	-	-	-	-	-	-
		GZSL	17.9	7.9	10.9	15.6	11.5	13.2	3.7	-	-	-	-	-	-	-
	LESA [8]	ZSL	25.7	41.1	31.6	19.7	52.5	28.7	19.4	0.7	25.6	1.4	0.5	37.4	1.0	41.7
		GZSL	23.6	10.4	14.4	19.8	14.6	16.8	5.6	16.2	18.9	17.4	10.2	23.9	14.3	45.4
	BiAM [10]	ZSL	-	-	33.1	-	-	30.7	26.3	-	-	8.3	-	-	5.5	62.2*
		GZSL	-	-	16.1	-	-	19.0	9.3	-	-	19.1	-	-	15.9	67.1*
	SDL [11]	ZSL	24.2	41.3	30.5	18.8	53.4	27.8	25.9	**6.1**	**47.0**	**10.7**	**4.4**	**68.1**	**8.3**	62.9
		GZSL	27.7	13.9	18.5	23.0	19.3	21.0	12.1	35.3	40.8	37.8	23.6	54.5	32.9	75.3
GEN	Gen-ADA [12]	ZSL	26.0	41.1	31.9	19.9	52.3	28.8	26.3	-	-	-	-	-	-	-
		GZSL	30.2	13.1	18.3	25.2	18.3	21.2	11.0	-	-	-	-	-	-	-
	Gen-MLZSL [9]	ZSL	26.6	42.8	32.8	20.1	53.6	29.3	25.7	1.3	42.4	2.5	1.1	52.1	2.2	43.0
		GZSL	30.9	13.6	18.9	26.0	19.1	22.0	8.9	33.6	38.9	36.1	22.8	52.8	31.9	49.7
	TGF [ours]	ZSL	**29.0**	**46.3**	**35.6**	**21.4**	**56.9**	**31.1**	**31.1**	4.0	31.0	7.1	3.0	47.1	5.7	**63.5**
		GZSL	**33.9**	**14.9**	**20.7**	**29.1**	**21.4**	**24.6**	**15.8**	**36.9**	**42.7**	**39.6**	**24.2**	**56.0**	**33.8**	**77.6**

TABLE II

STATE-OF-THE-ART COMPARISON FOR MLZSL AND MLGZSL TASKS ON THE MS COCO DATASET. WE REPORT THE RESULTS IN TERMS OF MAP, AS WELL AS PRECISION (P), RECALL (R), AND F1 SCORE AT $K \in \{3, 5\}$. THE BEST AND THE SECOND BEST RESULTS ARE MARKED IN BOLD AND RED, RESPECTIVELY

Methods	Task	MS COCO (#seen / #unseen = 65 / 15)						
		K=3			K=5			mAP
		P	R	F1	P	R	F1	
BiAM [10]	ZSL	27.6	77.4	40.7	-	-	-	37.7
	GZSL	39.3	36.2	37.7	31.3	48.0	37.9	37.2
SDL [11]	ZSL	**28.4**	**79.4**	**41.8**	-	-	-	38.0
	GZSL	44.2	40.7	42.4	36.7	56.3	44.4	39.9
Gen-MLZSL [9]	ZSL	26.5	74.2	39.1	-	-	-	46.9
	GZSL	47.6	43.8	45.6	37.0	56.8	44.8	36.4
TGF [Ours]	ZSL	27.8	77.9	41.0	-	-	-	**49.3**
	GZSL	**49.3**	**45.4**	**47.3**	**38.3**	**58.7**	**46.3**	**40.3**

images, 41,620 and 125,456 images in validation and test sets, respectively. As in [8], [9], [10], and [11], 7186 labels with at least 100 training images are selected as seen classes, and the 400 test set labels that appear least frequently in the training data are selected as the unseen classes.The **MS COCO** dataset is divided into a training set and a validation set, containing 82,783 and 40,504 images, respectively. This dataset is commonly used in multi-label zero-shot object detection [59], [60], here we conduct multi-label zero-shot classification experiments by using the same split (65 seen classes and 15 unseen classes) as in [59] and [60].

2) Evaluation Metrics: As in previous works [8], [9], [10], [11], we use F1 score at *top-K* predictions [43] and mean Average Precision (mAP) [61] as evaluation metrics. Specifically, the F1 score measures the model's ability to correctly rank the labels in each image, while the mAP score captures how accurately the model ranks the images for each label. In addition, since F1 score is the harmonic mean of the precision (P) and recall (R) (*i.e.*, $F1 = 2 * P * R/(P + R)$), so we also report the precision and recall in our experiments for a more detailed evaluation.

3) Implementation Details: Following previous works [8], [9], [10], [43], we employ the ℓ_2 normalized 300-dimensional GloVe [32] vector corresponding to the class name as the word vector w. The dimension of the noise z is equal to the dimension of the multi-label semantic embedding s (*i.e.*, 600). Both SEAs and FTN in our method are implemented as a multilayer perceptron (MLP) with one hidden layer. SEAs have 600 hidden units with LeakyReLU as non-linearity; FTN has 8192 hidden units with ReLU as non-linearity. For our FGNs, E_t, E_l, D_t, and D_l are all two-layer fully connected (FC) networks with 4096 hidden units and use LeakyReLU as nonlinear activation function. Both G_t and G_l consist of two FC+LeakyReLU layers with 4096 hidden units, and a residual branch (an MLP with one hidden layer and 8192 hidden units), and the output layers of G_t and G_l use Sigmoid as nonlinear activation function. FGNs is trained using the Adam optimizer with a learning rate of $1e^{-4}$ and a batch size of 64 for 35 epochs on NUS-WIDE and 5 epochs on Open Images. In addition, both MLZSL and MLZSL classifiers are trained

on NUS-WIDE and Open Images using (batch size, learning rate) of (256, $1e^{-3}$). Our TGF is implemented with PyTorch, and all the experiments are performed on an NVIDIA GeForce RTX 3090 GPU.

B. Comparing With the State-of-the-Art

Table I presents the performance comparison of the current state-of-the-art MLZSL methods. Among them, the non-generative methods [8], [10], [11], [43], [54], [55], [56] learn a multi-label classification model on the seen classes and directly transfer it for unseen class recognition. The generative methods [9], [12] regard MLZSL as an unseen data-missing problem, and transform MLZSL into a traditional supervised classification task by synthesizing unseen class samples.

As can be seen from Table I, our TGF outperforms all state-of-the-art methods (both in ZSL and GZSL tasks) on **NUS-WIDE** dataset with absolute gains of 4.8% (ZSL) and 3.7% (GZSL) in mAP, and in terms of F1 score at $K \in \{3, 5\}$ consistent improvements are also obtained. This demonstrates the effectiveness of our method.

Additionally, on **Open Images**, we achieve state-of-the-art mAP and F1 score for the GZSL task; and for the ZSL task, we also achieve the highest mAP and competitive F1 score (third best at $K = 10$ and second best at $K = 20$). Compared to NUS-WIDE, Open Images contains a significantly larger number of seen (7186) and unseen (400) classes and more training samples (close to 9 million), which makes the learning of SEAs, FTN, and FGNs in our TGF more challenging, so it is better to construct more complex network architectures for them. However, in this paper, both SEA and FTN use an MLP with only one hidden layer, and FGNs only uses a simple f-VAEGAN [15] as the generative backbone. We believe that for Open Images, the proposed TGF can integrate more complex network architectures to further improve the performance.

We also conduct experiments on the **MS COCO** dataset and compare the performance of our TGF with the previous best non-generative methods (*i.e.*, BIAM [10] and SDL [11]) and generative methods (*i.e.*, Gen-MLZSL [9]). Since an image contains at most three unseen classes in the validation set of MS COCO, we only report the F1 score at $K = 3$ for the ZSL task. From the results in Table II, we can see that our TGF achieves the highest mAP and F1 score at $K \in \{3, 5\}$ on the GZSL task. For the ZSL task, TGF also achieves the highest mAP with an absolute gain of 2.4%, while achieving the second-best F1 score at $K = 3$, only 0.8% lower than SDL.

Furthermore, our TGF shows great superiority compared to two existing generative MLZSL methods [9], [12], significantly improving F1 score and mAP in both ZSL and GZSL tasks. This provides further evidence for the effectiveness of TGF. It can generate discriminative multi-label semantic embeddings using SEAs, and transform original global features into an embedding space that is semantic-related and more suitable for multi-label classification using FTN, thus providing better semantic and visual inputs for FGNs. Meanwhile, FGNs can generate both global and local features to train the final classifiers, which is beneficial for recognizing both dominant and small categories in multi-label images. In summary, our TGF is state-of-the-art among all existing generative MLZSL methods and is a successful exploration of using generative models to tackle MLZSL task.

C. mAP Improvement Comparison

To demonstrate the effectiveness of our TGF, we perform Average Precision (AP) comparisons on each category with the previous state-of-the-art generative method Gen-MLZSL [9] on NUS-WIDE. Figure 6 shows the AP comparison of all 81 unseen classes, our TGF surpasses Gen-MLZSL on 67 classes. TGF achieves significant improvements (more than 20%) on several unseen labels such as '*protest*' and '*airport*', while has relatively smaller (less than 10%) negative effects on labels such as '*moon*' and '*wedding*'. In addition, TGF is especially better at recognizing some abstract concepts (*e.g.*, '*protest*', '*cityscape*', '*sunset*', '*town*', '*military*', '*nighttime*', and '*sports*'), which suggests that: **(1)** Our proposed SEAs effectively capture the correlations between labels in a multi-label image, it can generate discriminative multi-label semantic embeddings. **(2)** The proposed FTN enhances the visual-semantic consistency, which benefits our FGNs to generate semantic-related multi-label features to help the recognition of corresponding abstract semantic concepts. **(3)** In addition to global features, our proposed FGNs can also generate local features, which are beneficial to recognize the class labels of multiple small objects in a multi-label image, thereby recognizing abstract concepts according to their interdependencies.

D. Qualitative Results

Figure 7 shows a comparison between the *top-5* tags returned by the previous state-of-the-art generative method Gen-MLZSL [9] and our TGF for some unseen example images from NUS-WIDE. The tags in olive green color appear in ground-truth annotations; those in red color are the wrong tags. From Figure 7, we can see that our TGF significantly outperforms Gen-MLZSL. In detail, in Figure 7a, Gen-MLZSL only predicts the dominant class '*mountain*' which occupies a larger region of the image, while our TGF also recognizes the class '*dog*' which occupies a relatively smaller region of the image. In Figure 7d, Gen-MLZSL only recognizes the dominant categories '*house*' and '*sky*', while our TGF can also accurately predicts small object '*tree*'. In Figure 7e and Figure 7f, Gen-MLZSL can only recognize the dominant categories '*ocean*', '*sky*' and '*clouds*' in the image, while our TGF further successfully predicts the very small class '*boats*'. These qualitative results demonstrate the effectiveness of our TGF. It can synthesize both discriminative global and local features to exploit their ability to recognize the dominant and minor object in a multi-label image, respectively.

In addition, in Figure 7b, Gen-MLZSL only recognizes two dominant classes of '*sky*' and '*buildings*', while our TGF successfully predicts two abstract concepts of '*nighttime*' and '*cityscape*'. In Figure 7c, Gen-MLZSL only recognizes the dominant classes '*sky*', '*ocean*' and '*mountain*', while our TGF can also predict the abstract concept '*sunset*'.

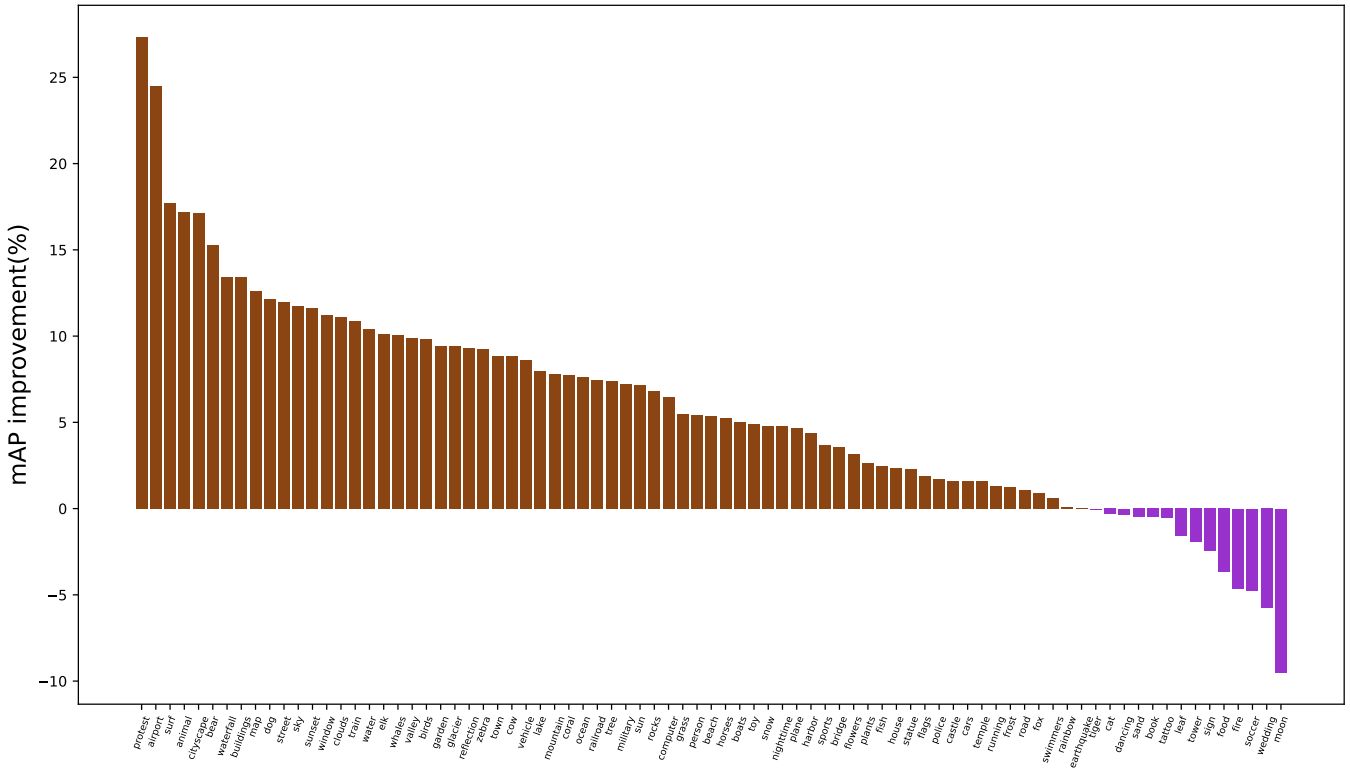


Fig. 6. Comparison of mAP improvement between our TGF and the previous state-of-the-art generative method Gen-MLZSL [9] on NUS-WIDE. Among all 81 unseen classes, our TGF outperforms Gen-MLZSL on 67 classes in terms of AP.

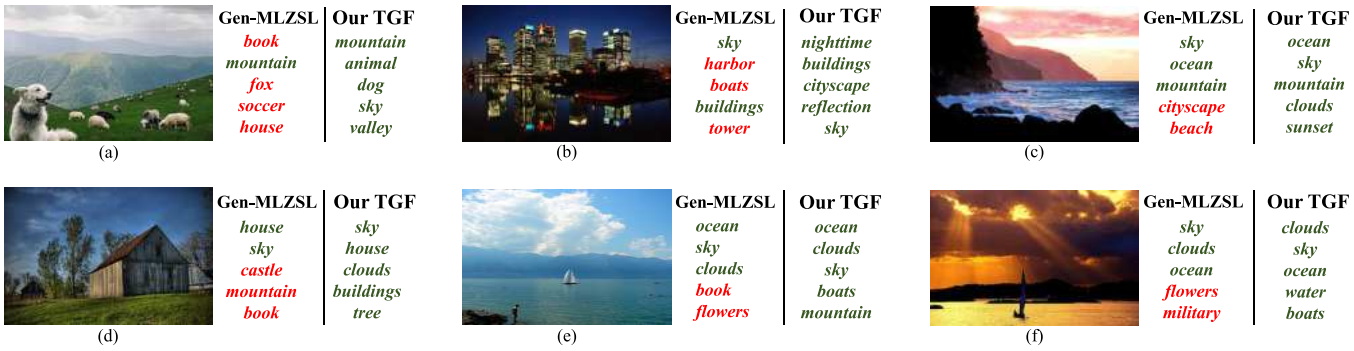


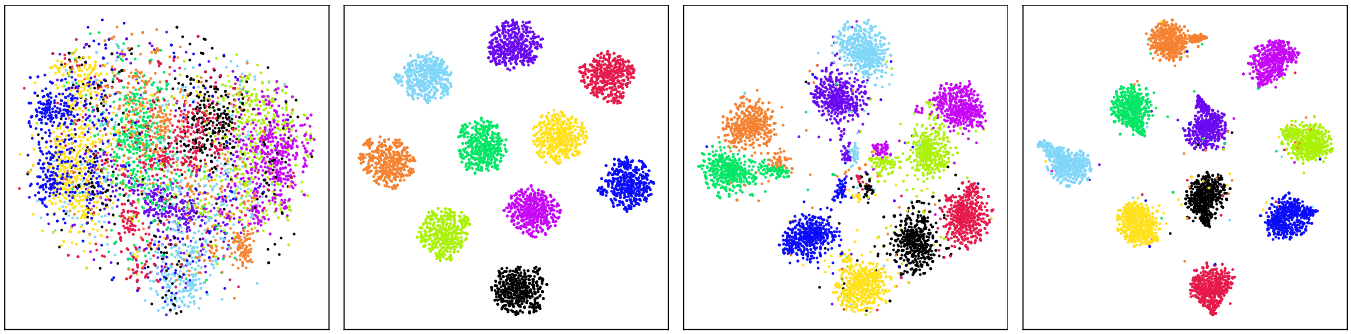
Fig. 7. Qualitative results on several unseen exemplar images from the NUS-WIDE test set. The *top-5* predictions per image for both methods are shown in ***olive green*** (true positives) and ***red*** (false positives). Best viewed in color.

This is consistent with the results in Section IV-C, further demonstrating the effectiveness of our TGF in recognizing abstract categories: **(a)** TGF utilizes SEAs to learn the correlation between labels in a multi-label image, thereby generating multi-label semantic embeddings to synthesize discriminative multi-label features. **(b)** TGF uses FTN to enhance the visual-semantic consistency, which facilitates subsequent FGNs to generate semantic-related multi-label features to help the recognition of corresponding abstract semantic concepts. **(c)** The local features generated by FGNs are beneficial for classifying multiple small objects in a multi-label image, thereby recognizing abstract concepts based on their interdependencies. Overall, the results demonstrate the superior performance of our proposed TGF over Gen-MLZSL.

E. t-SNE Visualization

To gain further insight into the quality of the multi-label semantic embeddings and visual features generated by our TGF, we visualize them on the NUS-WIDE dataset using t-SNE [62]. Since an image contains more than one category in a multi-label dataset, this prevents us from assigning a label to a sample for t-SNE visualization like in a single-label dataset. Therefore, we design new schemes for the visualization of multi-label semantic embeddings and multi-label features, the details are as follows.

1) Visualization of Multi-label Semantic Embeddings: We consider all 1006 categories (925 seen and 81 unseen) in NUS-WIDE as subcategories, and use K-means to cluster them into 50 parent categories. Then, we randomly divide



(a) Features generated by Gen-MLZSL (b) Multi-label semantic embeddings obtained by our SEAs (c) Local features generated by our FGNs (d) Global features generated by our FGNs

Fig. 8. t-SNE visualization on NUS-WIDE dataset. (a) Features generated by the previous state-of-the-art generative method Gen-MLZSL [9]. (b) Multi-label semantic embeddings obtained by our SEAs. (c) Local features synthesized by our FGNs. (d) Global features synthesized by our FGNs.

TABLE III
THE RESULTS OF USING DIFFERENT CNN BACKBONES ON NUS-WIDE. THE BEST RESULTS ARE MARKED IN BOLD

Methods	Task	K=3			K=5			mAP
		P	R	F1	P	R	F1	
Gen-MLZSL [9] (VGG19)	ZSL	26.2	41.9	32.3	20.0	53.4	29.1	25.2
	GZSL	30.8	13.6	18.9	25.8	19.0	21.9	8.7
TGF (VGG19)	ZSL	**27.4**	**43.8**	**33.7**	**20.4**	**54.2**	**29.6**	**28.4**
	GZSL	**31.7**	**14.0**	**19.4**	**27.0**	**19.8**	**22.9**	**13.0**
Gen-MLZSL [9] (Res101)	ZSL	26.6	42.6	32.8	20.0	53.3	29.1	25.9
	GZSL	32.1	14.2	19.6	26.8	19.7	22.7	9.9
TGF (Res101)	ZSL	**29.0**	**46.3**	**35.6**	**21.4**	**56.9**	**31.1**	**31.1**
	GZSL	**33.9**	**14.9**	**20.7**	**29.1**	**21.4**	**24.6**	**15.8**

these 50 parent categories into 10 groups. For each group, subcategories are randomly sampled from each parent category each time to form a multi-label combination, and each group is sampled 500 times. Next, we use SEAs to generate multi-label semantic embeddings for these multi-label combinations. Finally, we treat the multi-label combinations belonging to the same group as the same category and assign them the same label for t-SNE visualization. Here we give a simple example for easy understanding: suppose *Group_1* = {*Parent_category_1*, *Parent_category_2*, ..., *Parent_category_5*}, and *Parent_category_1* = {*dog*, *cat*, ..., *puppy*}, *Parent_category_2* = {*car*, *truck*, ..., *jeep*}, *Parent_category_3* = {*football*, *basketball*, ..., *soccer*}, *Parent_category_4* = {*dress*, *clothing*, ..., *shoes*}, *Parent_category_5* = {*girl*, *woman*, ..., *child*}. First, we sample from each parent category separately to construct multi-label combinations (*e.g.*, {*dog*, *car*, *basketball*, *dress*, *woman*}, {*cat*, *truck*, *soccer*, *shoes*, *child*}, {*puppy*, *car*, *football*, *clothing*, *girl*}, etc.). We then use SEAs to generate multi-label semantic embeddings for these multi-label combinations belonging to the *Group_1*, and assign them the same label for t-SNE visualization. The result of multi-label semantic embedding visualization is shown in Figure 8b. It can be seen that our SEAs aggregates multi-label semantic embeddings belonging to the same group and easily separates multi-label semantic embeddings belonging to different groups. This demonstrates that our SEAs can generate discriminative multi-label semantic embeddings that preserve semantic information belonging to multiple categories.

2) Visualization of Multi-label Visual Features: We randomly construct 10 multi-label combinations for the 81 unseen classes of NUS-WIDE, and use TGF to generate 500 multi-label features for each multi-label combination. Here we treat each multi-label combination as a class for t-SNE visualization as in single-label dataset. For example, *Multi-label_combination_1* = {*moon*, *animal*, *rocks*, *military*, *cat*}, we first use SEAs to generate multi-label semantic embeddings for this multi-label combination, and then use this multi-label semantic embedding as the conditional input of FGNs to generate 500 local features and 500 global features. Additionally, we also generate features for these 10 multi-label combinations using the previous state-of-the-art generative method Gen-MLZSL [9]. It can be seen from Figure 8a that for different multi-label combinations, the feature distributions generated by Gen-MLZSL are highly overlapping, while our TGF generated local features (Figure 8c) and global features (Figure 8b) both have good intra-class compactness and inter-class separation, which further demonstrates the effectiveness of our TGF. It is worth mentioning that compared with global features, the generated local features are more diverse and contain more noise, which is consistent with our conjecture in Section III-E: local features contain more image details and are more beneficial to recognize small object categories in a multi-label image, while global features are more discriminative and generalizable, and it focuses on dominant object categories that occupy larger areas of the image.

F. Comparison of Different CNN Backbones

Previous works [7], [8], [9], [10], [11], [12] all use VGG [49] or ResNet [50] pre-trained on Imagenet [51] as feature extractors. To eliminate the influence of different CNN backbones and ensure a fair comparison, we employ VGG19 and Res101 to extract image features respectively, and compare the performance of our TGF with the previous state-of-the-art generative MLZSL method Gen-MLZSL [9]. As shown in Table III, no matter using VGG19 or Res101 as the feature extraction network, our TGF significantly outperforms Gen-MLZSL, which further demonstrates the effectiveness of our method. The inferior performance of Gen-MLZSL can be attributed to the fact that ALF and FLF

TABLE IV
COMPARISON OF mAP WITH DIFFERENT CLASS EMBEDDINGS ON NUS-WIDE. THE BEST RESULTS ARE MARKED IN BOLD

Class Embeddings	ZSL	GZSL
Syntactic Embedding (SynGCN [63])	24.3	15.3
Semantic Embedding (GloVe [32])	31.1	15.8
Vision-Language Pre-training (CLIP [66])	**34.8**	**17.0**

TABLE V
THE RESULTS OF ABLATION STUDY FOR OBJECTIVE FUNCTIONS OF SEAs ON NUS-WIDE. THE BEST RESULTS ARE MARKED IN BOLD

$\mathcal{L}_{sim}$	$\mathcal{L}_{avg}$	$\mathcal{L}_{rec}$	K=3			K=5			mAP
			P	R	F1	P	R	F1	
✔	✗	✗	28.5	45.5	35.0	20.9	55.6	30.3	30.7
✔	✔	✗	28.7	45.9	35.3	21.1	56.3	30.7	30.8
✔	✔	✔	**29.0**	**46.3**	**35.6**	**21.4**	**56.9**	**31.1**	**31.1**

TABLE VI
THE RESULTS OF ABLATION STUDY FOR COMPONENTS IN TGF ON NUS-WIDE. THE BEST RESULTS ARE MARKED IN BOLD

Models	K=3			K=5			mAP
	P	R	F1	P	R	F1	
FGN-t + ALF	25.6	41.0	31.5	19.5	51.9	28.3	25.5
FGN-t + FLF	25.7	41.1	31.6	19.5	51.9	28.3	27.2
FGN-t + SEAs	26.5	42.3	32.6	19.9	53.0	28.9	29.0
FGN-t + SEAs + FTN	27.3	43.6	33.6	20.4	54.4	29.7	**31.8**
FGN-t + SEAs + FTN + FGN-l	**29.0**	**46.3**	**35.6**	**21.4**	**56.9**	**31.1**	31.1

destroy the structure and distribution of the original data due to simple averaging, and inevitably lose some discriminative information. In addition, Gen-MLZSL also requires a complex attention module to fuse the features output by ALF and FLF, which is computationally expensive and requires a significant memory footprint. In contrast, our TGF generates discriminative multi-label semantic embeddings via SEAs, using FTN transforms global features extracted from a CNN pre-trained on single-label images into semantic-related and more suitable features for multi-label classification. Furthermore, TGF also utilizes FGNs to synthesize discriminative global and local features to exploit their ability to recognize dominant and minor objects in a multi-label image, respectively.

G. The Impact of Different Class Embeddings

For a fair comparison with previous MLZSL methods [8], [9], [10], [43], we use the GloVe model [32] trained on Wikipedia articles to extract semantic word vectors as class embeddings. Here we study the effect of using different class embeddings on the performance of TGF. In Natural Language Processing (NLP), there are two approaches to representing words or phrases as numerical vectors: syntactic embedding and semantic embedding. **Syntactic embedding methods** [63], [64] focus on capturing the grammatical or syntactic structure of words or phrases, which represent words or phrases based on their syntactic relationships and dependencies. In a syntactic embedding model, words like '*tiger*' and '*chase*' may have similar vector representations because they often appear together in subject-verb relationships. **Semantic embedding methods** [32], [65] aim to capture the semantics of words or phrases, which represent words or phrases based on semantic similarity and relatedness. In a semantic embedding model, words like '*tiger*' and '*elephant*' would have similar vector representations because they semantically share similar properties, *i.e.*, both are animals. Furthermore, recently popular **Vision-language pre-training (VLP) models** such as CLIP [66] are trained on billions of image-text pairs and have a powerful image-text matching ability.

We use the syntactic embedding model SynGCN [63], the semantic embedding model Glove [32] and the text encoder of VLP-based model CLIP [66] to extract class embeddings for experiments. From Table IV, we can see that SynGCN has the worst performance because it fails to learn the semantic similarity and correlation between categories. In addition, although GloVe significantly enhances the performance compared to SynGCN by capturing the semantics of categories, it is trained only with text such as Wikipedia articles, so the obtained class semantic embeddings are poorly correlated with the corresponding visual features. Finally, CLIP-based class embeddings achieve the best performance on both ZSL and GZSL tasks. This indicates that CLIP-based class embeddings have better consistency with the corresponding categories of visual features, and our TGF can further improve performance by utilizing better class embeddings.

H. Ablation Studies and Hyper-Parameters Sensitivity Analysis

1) Ablation Studies for Objective Functions of SEAs: In this section, we conduct ablation studies for three objective functions (*i.e.*, $\mathcal{L}_{sim}$, $\mathcal{L}_{avg}$ and $\mathcal{L}_{rec}$) in the proposed SEAs to evaluate their respective importance. Table V presents the MLZSL results using different objective functions on NUS-WIDE. As can be seen in Table V, competitive performance is achieved using only $\mathcal{L}_{sim}$, which shows that with $\mathcal{L}_{sim}$, we learn the semantic similarity between s_{sim} and all word vectors corresponding to a multi-label image, thus preserving the class-specific semantic information. When adding $\mathcal{L}_{avg}$, both the F1 score and mAP achieved further improvement. This is because by optimizing $\mathcal{L}_{avg}$, s_{avg} learns the correlation between categories in a multi-label image. Finally, the combination of $\mathcal{L}sim$, $\mathcal{L}avg$, and $\mathcal{L}_{rec}$ results in the highest MLZSL classification accuracy, which indicates that these three objective functions work together to enhance each other to generate discriminative multi-label semantic embeddings.

2) Ablation Studies for Components in TGF: To better assess the contribution of each component in TGF, we perform an ablation study as shown in Table VI. In order to be able to use the generative model to solve the MLZSL task, Gen-MLZSL [9] proposes ALF (*i.e.*, averaging all word vectors corresponding to each multi-label image at the input of the generator) and FLF (*i.e.*, averaging the single-label features generated by all word vectors corresponding to each multi-label image at the output of the generator) to obtain a

TABLE VII
COMPARISON OF SEAS AND FTNS USING ONE AND TWO HIDDEN LAYERS ON NUS-WIDE. THE BEST RESULTS ARE MARKED IN BOLD

Models	ZSL	GZSL
One hidden layer	31.1	15.8
Two hidden layers	**31.6**	**16.4**

uniform input and output size. From the results in table VI, it can be observed that training FGN-t with the multi-label semantic embeddings s obtained from SEAs and the original global features x_g (*i.e.*, FGN-t + SEAs) outperforms FGN-t + ALF and FGN-t + FLF. This demonstrates that simple averaging in ALF and FLF destroys the original data structure and distribution (especially for samples containing a large number of classes), and inevitably loses some discriminative information, which is suboptimal for MLZSL. It also verifies that our proposed SEAs can generate discriminative multi-label semantic embeddings, enabling generative models to be trained to generate multi-label features for unseen classes.

When replacing x_g with semantic-related multi-label features x_t obtained from FTN (*i.e.*, FGN-t + SEAs + FTN), we achieve further improvements in both mAP and F1 score for MLZSL tasks. This demonstrates that our FTN enhances semantic-visual consistency and generates more discriminative multi-label features. Finally, adding local features x_l to train FGN-l (*i.e.*, FGN-t + SEAs + FTN + FGN+l) significantly improves the F1 score, while mAP only slightly decreases compared to SEAs+FTN+FGN-t. This validates our conjecture that local features are helpful for recognizing small objects in multi-label images, and it improves the model's ability to correctly rank all labels for each image. In addition, compared with the more discriminative and generalizable global features, local features contain more image details and thus more noise. The mAP, which reflects the ranking accuracy of all images for each label, is more sensitive to noise than the F1 score. Therefore, the mAP decreases slightly after adding local features.

3) FTN and SEAs With More Complex Network Architectures: In this paper, both SEA and FTN in our TGF are implemented as an MLP with only one hidden layer, and this simple network architecture has been able to achieve state-of-the-art performance on three MLZSL benchmark datasets. This suggests that our novel idea is the driving force behind the observed performance enhancement rather than complex network architecture. In order to investigate the impact of network complexity on TGF, this section explores the effect of increasing the number of hidden layers of both SEA and FTN from one to two layers. The outcomes of this analysis, as illustrated in Table VII, indicate that as the number of network layers increases, TGF achieves mAP improvement on both MLZSL and MLGZSL tasks. These findings confirm that more complex network architectures have the potential to further enhance the performance of TGF.

4) The Impact of Different N for SEAs: Hyper-parameter N determines how many SEA we need to learn. As shown in Figure 9a, when $N = 20$, we obtained the highest F1 score at both $K = 3$ and $K = 5$ for the MLZSL and MLGZSL tasks. As mentioned in Section III-C, for NUS-WIDE, $N = 20$ already encompasses 99% of the training set samples. We think that $N = 20$ is suitable for model training and also in line with real-world scenarios. Because N is too small (such as 5) to encompass an adequate number of training samples, too large N (such as 50) implies that one sample contains too many classes, which is very rare in both datasets and realistic scenarios, and can be seen as abnormal data that is detrimental for training. Therefore, we set $N = 20$, $N = 15$ and $N = 8$ for NUS-WIDE, Open Images and MS COCO, respectively.

5) The Effect of Hyper-Parameters α and β: α and β are the hyper-parameters that control the C-VAE and C-WGAN loss weights and the FGN-t and FGN-l loss weights in FGNs, respectively. As can be seen in Figure 9b, when $\alpha = 10$, we achieve the best performance for both MLZSL and MLGZSL tasks. This indicates that the weight of C-WGAN is preferably slightly larger than that of C-VAE. Moreover, Figure 9c shows that the variation of β has almost no effect on the performance, which indicates that global and local features are equally important for multi-label classification. Global features tend to recognize the dominant categories, while local features help us to recognize smaller categories in a multi-label image. Overall, the variation of both hyper-parameters α and β have little impact on performance, which demonstrates the robustness of our TGF.

I. Open-Vocabulary Multi-Label Classification

The recently developed Open-vocabulary object detection (OVOD) [67], [68], [69], [70] and Open-vocabulary semantic segmentation (OVSS) methods [71], [72], [73], [74] leverage the multi-modal knowledge of image-text pairs in Vision-language pre-training (VLP) models and achieve impressive performance. Inspired by these methods, [75] proposes Open-vocabulary multi-label classification (OVML), which aims to recognize multiple categories described by arbitrary text in a multi-label image. The difference between OVML and classic MLZSL is that the label embedding (*e.g.*, GloVe [32]) used in classic MLZSL can only handle word labels (*e.g.*, the label of '*fox*') well, while OVML can be easily extended to text labels (*e.g.*, the label of '*white fox*') by jointly exploring the multi-modal knowledge of the VLP model. References [75] proposes an OVML framework called Multi-modal knowledge transfer (MKT). MKT uses knowledge distillation to transfer the image-text matching ability of the pre-trained CLIP model, and performs prompt tuning to further update the label embedding. To further recognize multiple objects in a multi-label image, it also develops a dual-stream module to capture both local and global features. References [75] also propose an OVML baseline CLIP-FT, which is a pre-trained CLIP model [66] fine-tuned on seen categories according to ranking loss. In order to evaluate the ability of our TGF to utilize the multi-modal knowledge of the VLP model for OVML, we directly use the pre-trained CLIP image encoder (ViT-B/16, the same as MKT for fair comparison) and text encoder to extract global image features and class semantic embeddings

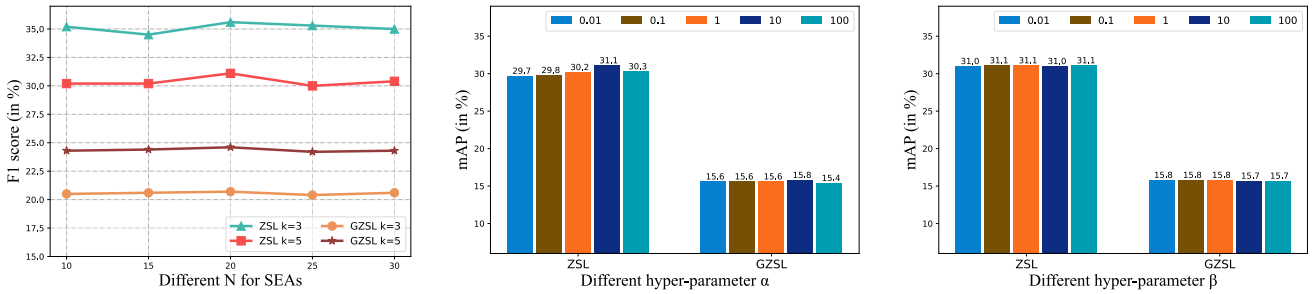


Fig. 9. The impact of different hyper-parameters N, α, β on NUS-WIDE.

TABLE VIII

STATE-OF-THE-ART COMPARISON FOR MLZSL AND MLGZSL TASKS IN OPEN-VOCABULARY (OV) SETTING ON THE NUS-WIDE DATASET. WE REPORT THE RESULTS IN TERMS OF mAP, AS WELL AS PRECISION (P), RECALL (R), AND F1 SCORE AT $K \in \{3, 5\}$. THE BEST RESULTS ARE MARKED IN BOLD

Methods	Setting	Task	K=3			K=5			mAP
			P	R	F1	P	R	F1	
TGF without pre-trained CLIP [ours]	ZS	ZSL	**29.0**	**46.3**	**35.6**	21.4	56.9	31.1	31.1
		GZSL	33.9	14.9	20.7	29.1	21.4	24.6	15.8
CLIP-FT [75]	OV	ZSL	19.1	30.5	23.5	14.9	39.7	21.7	30.5
		GZSL	33.2	14.6	20.3	27.4	20.2	23.2	16.8
MKT with pre-trained CLIP [75]		ZSL	27.7	44.3	34.1	21.4	57.0	31.1	37.6
		GZSL	35.9	15.8	22.0	29.9	22.0	25.4	18.3
TGF with pre-trained CLIP [ours]		ZSL	28.2	45.1	34.7	**22.0**	**58.5**	**31.9**	**37.6**
		GZSL	**37.8**	**16.6**	**23.1**	**30.6**	**22.5**	**26.0**	**22.6**

respectively to train our TGF, and the results are shown in Table VIII. It is worth noting that here we only use FGN-t + SEAs for experiments, without prompt tuning like MKT and without using additional local features. However, this simple integration of TGF with pre-trained CLIP has surpassed the state-of-the-art OVML method MKT, which further proves the effectiveness of our TGF, and also demonstrates that generative methods are a better choice to solve the unseen class samples missing problem (such as MLZSL and OVML).

V. CONCLUSION

This paper proposes a novel TGF for multi-label zero-shot learning that combines the advantages of SEAs, FTN, and FGNs. SEAs can encode class-level word vectors into sample-level multi-label semantic embedding, which fundamentally solves the limitations of generative models applied to MLZSL, thus enabling the transfer of arbitrarily existing generative SLZSL models to solve the MLZSL task. In addition, FTN further improves the performance of MLZSL by transforming the global features into a semantic-related and more suitable embedding space for multi-label classification. Finally, FGNs can generate both global and local features to exploit their ability to recognize dominant and small categories in a multi-label image, respectively. Extensive experimental results and analyses demonstrate the effectiveness of our TGF. Overall, TGF sets a new state-of-the-art on three multi-label zero-shot datasets, achieving absolute gains of up to 20.5% for ZSL and 27.9% for GZSL in terms of mAP compared to the previous best generative MLZSL method. It is worth mentioning that in this paper, TGF utilizes very simple architectures, and several components (SEAs, FTN, and FGNs) in it can be replaced by more complex network architectures to further improve the performance. TGF provides some inspiration for the application of generative models in MLZSL and MLGZSL.

REFERENCES

[1] L. Zang, Y. Li, and H. Chen, "Multilabel recognition algorithm with multigraph structure," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 2, pp. 782–792, Feb. 2023.

[2] Z. Wang, Z. Fang, D. Li, H. Yang, and W. Du, "Semantic supplementary network with prior information for multi-label image classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 1848–1859, Apr. 2022.

[3] T. Ridnik et al., "Asymmetric loss for multi-label classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 82–91.

[4] K. Zhu and J. Wu, "Residual attention: A simple but effective method for multi-label recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 184–193.

[5] Y. Wang et al., "Multi-label classification with label graph superimposing," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 12265–12272.

[6] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5177–5186.

[7] C.-W. Lee, W. Fang, C.-K. Yeh, and Y. F. Wang, "Multi-label zero-shot learning with structured knowledge graphs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1576–1585.

[8] D. Huynh and E. Elhamifar, "A shared multi-attention framework for multi-label zero-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8776–8786.

[9] A. Gupta, S. Narayan, S. Khan, F. S. Khan, L. Shao, and J. van de Weijer, "Generative multi-label zero-shot learning," 2021, *arXiv:2101.11606*.

[10] S. Narayan, A. Gupta, S. Khan, F. S. Khan, L. Shao, and M. Shah, "Discriminative region-based multi-label zero-shot learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8731–8740.

[11] A. Ben-Cohen, N. Zamir, E. B. Baruch, I. Friedman, and L. Zelnik-Manor, "Semantic diversity learning for zero-shot multi-label classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 640–650.

[12] K.-Y. Chen and M.-C. Yeh, "Generative and adaptive multi-label generalized zero-shot learning," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2022, pp. 1–6.

[13] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning—A comprehensive evaluation of the good, the bad and the ugly," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2251–2265, Sep. 2019.

[14] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: A real-world web image database from National University of Singapore," in *Proc. ACM Int. Conf. Image Video Retr.*, Jul. 2009, pp. 1–9.

[15] Y. Xian, S. Sharma, B. Schiele, and Z. Akata, "F-VAEGAN-d2: A feature generating framework for any-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10275–10284.

[16] J. Li, M. Jing, K. Lu, Z. Ding, L. Zhu, and Z. Huang, "Leveraging the invariant side of generative zero-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 7402–7411.

[17] P. Ma and X. Hu, "A variational autoencoder with deep embedding model for generalized zero-shot learning," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 11733–11740.

[18] S. Narayan, A. Gupta, F. S. Khan, C. G. M. Snoek, and L. Shao, "Latent embedding feedback and discriminative features for zero-shot classification," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 479–495.

[19] S. Chen et al., "FREE: Feature refinement for generalized zero-shot learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 122–131.

[20] Z. Han, Z. Fu, S. Chen, and J. Yang, "Contrastive embedding for generalized zero-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2371–2381.

[21] Z. Chen et al., "Semantics disentangling for generalized zero-shot learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8712–8720.

[22] Z. Wang, Y. Gou, J. Li, L. Zhu, and H. T. Shen, "Language-augmented pixel embedding for generalized zero-shot learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 3, pp. 1019–1030, Mar. 2023.

[23] S. Chen et al., "GNDAN: Graph navigated dual attention network for zero-shot learning," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, May 4, 2022, doi: 10.1109/TNNLS.2022.3155602.

[24] H. Su, J. Li, K. Lu, L. Zhu, and H. T. Shen, "Dual-aligned feature confusion alleviation for generalized zero-shot learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 8, pp. 3774–3785, Aug. 2023.

[25] D. Cheng, G. Wang, N. Wang, D. Zhang, Q. Zhang, and X. Gao, "Discriminative and robust attribute alignment for zero-shot learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 8, pp. 4244–4256, Aug. 2023.

[26] Y. Li, Z. Liu, L. Yao, X. Wang, J. McAuley, and X. Chang, "An entropy-guided reinforced partial convolutional network for zero-shot learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 8, pp. 5175–5186, Aug. 2022.

[27] J. Shen, Z. Xiao, X. Zhen, and L. Zhang, "Spherical zero-shot learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 2, pp. 634–645, Feb. 2022.

[28] L. Zhang et al., "Towards effective deep embedding for zero-shot learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 9, pp. 2843–2852, Sep. 2020.

[29] V. Ferrari and A. Zisserman, "Learning visual attributes," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 20, 2007, pp. 433–440.

[30] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1778–1785.

[31] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 3111–3119.

[32] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[33] S. Reed, Z. Akata, H. Lee, and B. Schiele, "Learning deep representations of fine-grained visual descriptions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 49–58.

[34] A. Frome et al., "DeViSE: A deep visual-semantic embedding model," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 2121–2129.

[35] B. Romera-Paredes and P. Torr, "An embarrassingly simple approach to zero-shot learning," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2152–2161.

[36] L. Zhang, T. Xiang, and S. Gong, "Learning a deep embedding model for zero-shot learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2021–2030.

[37] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1199–1208.

[38] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong, "Transductive multi-view zero-shot learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 11, pp. 2332–2345, Nov. 2015.

[39] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 2672–2680.

[40] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[41] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.

[42] F. Zhou, S. Huang, B. Liu, and D. Yang, "Multi-label image classification via category prototype compositional learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 7, pp. 4513–4525, Jul. 2022.

[43] Y. Zhang, B. Gong, and M. Shah, "Fast zero-shot image tagging," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5985–5994.

[44] L. Wang, Z. Ding, S. Han, J.-J. Han, C. Choi, and Y. Fu, "Generative correlation discovery network for multi-label learning," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2019, pp. 588–597.

[45] L. Wang et al., "Generative multi-label correlation learning," *ACM Trans. Knowl. Discovery Data*, vol. 17, no. 2, pp. 1–19, Apr. 2023.

[46] G. Patterson, C. Xu, H. Su, and J. Hays, "The SUN attribute database: Beyond categories for deeper scene understanding," *Int. J. Comput. Vis.*, vol. 108, nos. 1–2, pp. 59–81, May 2014.

[47] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD birds-200–2011 dataset," California Inst. Technol., California, Tech. Rep. CNS-TR-2011-001, 2011.

[48] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 453–465, Mar. 2014.

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

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[51] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[52] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2013, pp. 818–833.

[53] W. Zhai et al., "An attentive hierarchy ConvNet for crowd counting in smart city," *Cluster Comput.*, vol. 26, no. 2, pp. 1099–1111, Apr. 2023.

[54] M. Norouzi et al., "Zero-shot learning by convex combination of semantic embeddings," in *Proc. Int. Conf. Learn. Represent.*, 2013, pp. 1–9.

[55] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-embedding for image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1425–1438, Jul. 2016.

[56] J.-H. Kim, J. Jun, and B.-T. Zhang, "Bilinear attention networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–11.

[57] I. Krasin et al. (2017). *OpenImages: A Public Dataset for Large-Scale Multi-Label and Multi-Class Image Classification*. [Online]. Available: https://github.com/openimages

[58] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Computer Vision—ECCV 2014*. Zürich, Switzerland: Springer, Sep. 2014, pp. 740–755.

[59] S. Rahman, S. Khan, and N. Barnes, "Improved visual-semantic alignment for zero-shot object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 11932–11939.

[60] N. Hayat, M. Hayat, S. Rahman, S. Khan, S. W. Zamir, and F. S. Khan, "Synthesizing the unseen for zero-shot object detection," in *Proc. Asian Conf. Comput. Vis.*, 2020, pp. 1–16.
[61] A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, and S. Belongie, "Learning from noisy large-scale datasets with minimal supervision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 839–847.
[62] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
[63] S. Vashishth, M. Bhandari, P. Yadav, P. Rai, C. Bhattacharyya, and P. P. Talukdar, "Incorporating syntactic and semantic information in word embeddings using graph convolutional networks," in *Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 3308–3318.
[64] O. Levy and Y. Goldberg, "Dependency-based word embeddings," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, 2014, pp. 302–308.
[65] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. Int. Conf. Learn. Represent.*, 2013, pp. 1–12.
[66] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.
[67] S. Zhao et al., "Exploiting unlabeled data with vision and language models for object detection," in *Proc. 17th Eur. Conf. Comput. Vis.*, Tel Aviv, Israel, Oct. 2022, pp. 159–175.
[68] C. Feng et al., "PromptDet: Towards open-vocabulary detection using uncurated images," in *Computer Vision—ECCV 2022*. Tel Aviv, Israel: Springer, Oct. 2022, pp. 701–717.
[69] Y. Zhong et al., "RegionCLIP: Region-based language-image pretraining," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 16793–16803.
[70] Y. Du, F. Wei, Z. Zhang, M. Shi, Y. Gao, and G. Li, "Learning to prompt for open-vocabulary object detection with vision-language model," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14084–14093.
[71] M. Xu et al., "A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model," in *Proc. 17th Eur. Conf. Comput. Vis.*, Tel Aviv, Israel, Oct. 2022, pp. 736–753.
[72] F. Liang et al., "Open-vocabulary semantic segmentation with mask-adapted clip," 2022, *arXiv:2210.04150*.
[73] H. Luo, J. Bao, Y. Wu, X. He, and T. Li, "SegCLIP: Patch aggregation with learnable centers for open-vocabulary semantic segmentation," 2022, *arXiv:2211.14813*.
[74] M. Xu, Z. Zhang, F. Wei, H. Hu, and X. Bai, "Side adapter network for open-vocabulary semantic segmentation," 2023, *arXiv:2302.12242*.
[75] S. He, T. Guo, T. Dai, R. Qiao, B. Ren, and S.-T. Xia, "Open-vocabulary multi-label classification via multi-modal knowledge transfer," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 808–816.

Peirong Ma received the M.S. degree in electronics and communication engineering from Guangzhou University, Guangzhou, China, in 2020. He is currently pursuing the Ph.D. degree with the School of Computer Science, Fudan University, Shanghai, China. His current research interests include zero-shot learning, open-set learning, image classification, and object detection.

Zhiquan He received the B.S. degree in software engineering from Fudan University, Shanghai, China, in 2022, where he is currently pursuing the M.S. degree with the Shanghai Key Laboratory of Intelligent Information Processing, School of Computer Science. His current research interests include image processing, image restoration, and computer vision.

Wu Ran (Graduate Student Member, IEEE) received the bachelor's degree in physics from Fudan University, Shanghai, China, in 2019, where he is currently pursuing the Ph.D. degree in computer application technology (CAT). His current research interests include deep learning-based image restoration (such as image de-raining and image denoise), meta-learning, and traditional machine learning.

Hong Lu (Member, IEEE) received the B.Eng. and M.Eng. degrees in computer science and technology from Xidian University, Xi'an, China, in 1993 and 1998, respectively, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2005. From 1993 to 2000, she was a Lecturer and a Researcher with the School of Computer Science and Technology, Xidian University. From 2000 to 2003, she was a Research Student with the School of Electrical and Electronic Engineering, Nanyang Technological University. Since 2004, she has been with the School of Computer Science, Fudan University, Shanghai, China, where she is currently a Professor. Her current research interests include computer vision, machine learning, pattern recognition, and robotic tasks.