A Joint Generative Model for Zero-Shot Learning

Rui Gao[1], Xingsong Hou[1], Jie Qin[2], Li Liu[3], Fan Zhu[3], and Zhao Zhang[4]

[1] School of Electronic and Information Engineering, Xi'an Jiaotong University, China
[2] Computer Vision Laboratory, ETH Zurich, Switzerland
[3] Inception Institute of Artificial Intelligence, UAE
[4] Soochow University, China

Abstract. Zero-shot learning (ZSL) is a challenging task due to the lack of data from unseen classes during training. Existing methods tend to have the strong bias towards seen classes, which is also known as the domain shift problem. To mitigate the gap between seen and unseen class data, we propose a joint generative model to synthesize features as the replacement for unseen data. Based on the generated features, the conventional ZSL problem can be tackled in a supervised way. Specifically, our framework integrates Variational Autoencoders (VAE) and Generative Adversarial Networks (GAN) conditioned on class-level semantic attributes for feature generation based on element-wise and holistic reconstruction. A categorization network acts as the additional guide to generate features beneficial for the subsequent classification task. Moreover, we propose a perceptual reconstruction loss to preserve semantic similarities. Experimental results on five benchmarks show the superiority of our framework over the state-of-the-art approaches in terms of both conventional ZSL and generalized ZSL settings.

Keywords: Zero-shot learning · variational autoencoder · generative adversarial network · perceptual reconstruction.

1 Introduction

Deep learning contributes significantly to the rapid progress in computer vision owing to its strong capabilities of data representation. However, there exists a non-negligible issue that training deep neural networks requires a huge amount of annotated data, which is usually unavailable in realistic scenarios due to labor-intensive data annotations. Meanwhile, with the explosive growth of new categories (*e.g.* of objects), it is even impossible to get any training data from certain classes. To deal with this, zero-shot learning (ZSL) has recently emerged as an effective solution [17, 19, 28, 18]. ZSL considers a more challenging case that training (seen) and test (unseen) classes are disjoint, *i.e.* the data of unseen classes is totally missing during the training process.

Specific intermediate representations (*e.g.* semantic attributes [18, 8, 11] and word vectors [37, 27, 48, 10]) have been widely used by ZSL methods to bridge

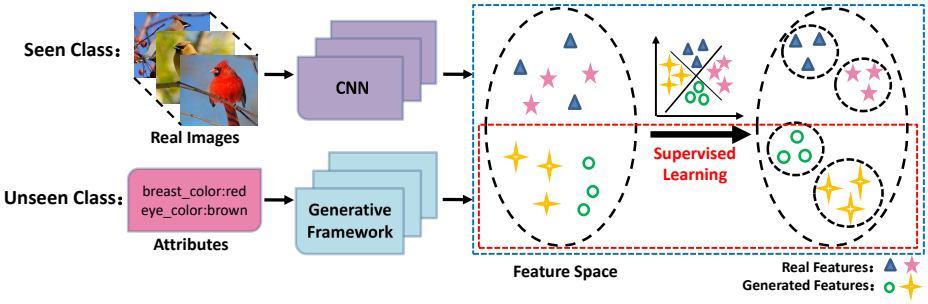


Fig. 1. The flow chart of the proposed approach. We address the zero-shot learning (ZSL) problem in a supervised way, by generating features for unseen classes via our generative framework. The red dotted box denotes the conventional ZSL task, and the blue dotted box denotes the generalized ZSL (GZSL) task.

the gap between seen and unseen classes. However, an inherent problem, known as 'domain shift' [12], still remains challenging for conventional ZSL methods. In other words, classifiers trained on seen classes are not suitable for unseen ones due to their different underlying distributions. Consequently, most existing methods have the strong bias towards seen data and their performance is unacceptable for conventional ZSL settings, let alone the recently proposed more realistic generalized ZSL (GZSL) settings [7, 42] where both seen and unseen classes are present at test time. Therefore, it is highly desirable to develop a generalized framework that could mitigate the domain shift and provide a universal classifier for both seen and unseen classes. As shown in Figure 1, in this work, we aim to address the above issues from a new perspective, *i.e.* converting ZSL to supervised learning, by hallucinating unseen class features based on deep generative models.

Deep generative models, such as Generative Adversarial Networks (GAN) and Variational Autoencoders (VAE), have been extensively studied in the recent few years. GAN [13] is appealing to generate realistic images, especially conditioned on additional information [33, 26]. VAE [15], especially the conditional VAE (CVAE) [38], has great potential to generate data through element-wise similarity metrics. In a similar spirit to our work, Xian *et al.* [43] proposed a ZSL framework to generate features for unseen classes based on conditional GAN (CGAN). However, GAN generally concentrates on more abstract and global data structure. In our problem, element-wise reconstruction is also essential for hallucinating unseen classes. Thus, we propose a joint framework by taking the advantages of CGAN and CVAE for more delicate data generation. Note that existing works [21, 4] have already shown the effectiveness of this kind of generative model in image synthesis. In contrast, we aim at generating features instead of images for unseen classes since the generated images are typically of insufficient quality to train deep networks for the final classification [43]. We add an additional categorization network to ensure the discriminability of the

synthesized features. Different from [43], the categorizer and the generator in our framework compete in a two-player minimax game. That is, the generator tries to generate features that belong to the classes of real features, and the categorizer tries to distinguish the generated features from the real ones in the category level simultaneously. Through the competition, the generated features will be well suited for training the final discriminative classifier. Moreover, we propose a perceptual reconstruction loss to preserve class-wise semantics based on the intermediate outputs of the discriminator and the categorizer.

The main contributions of this paper are summarized as follows:

- We propose a novel generative framework for zero-shot learning, which addresses conventional ZSL problems in a supervised manner. The framework takes the advantages of CGAN and CVAE to generate features conditioned on semantic attributes with the additional help of a categorization network. As a result, the generated features are not only similar to the real ones but also discriminative for the subsequent classification task.
- We leverage the intermediate outputs of the networks for perceptual reconstruction so that the generated features have the pixel-wise similarity as well as the semantic similarity to the real features.
- Extensive experimental results on five standard ZSL benchmarks demonstrate that the proposed method achieves notable improvement over the state-of-the-art approaches in not only the conventional ZSL but also the more challenging GZSL tasks.

The remainder of the paper is organized as follows. In Section 2, we give a brief review of existing ZSL methods and generative models. In Section 3, we introduce the proposed joint generative framework, which includes several networks to synthesize high-quality features of unseen classes for the subsequent classification task. Section 4 first introduces the datasets and experimental setup and then provides the demonstration of the experimental results. We finally draw our conclusion in Section 5.

2 Related Work

2.1 Zero-Shot Learning

Zero-shot learning is a challenging task because of the lack of training data. Many attempts [17, 19, 28, 18, 8, 27, 48, 1, 31, 34, 45, 30] have been made to exploit the relationships between seen and unseen classes. Semantic representations, such as semantic attributes [17, 18, 8, 11, 9] and word vectors [37, 27, 48, 10], are employed as the intermediate embedding to bridge the gap between the visual space and class space. Typically, a mapping from the visual space to semantic space is learned and then leveraged for the following classification task.

Recently, there were some works that learned the inverse mapping from the semantic space to visual space [43, 24, 23, 46, 5], which was shown effective for mitigating the domain shift problem. For instance, Zhang et al. [46] proposed

an end-to-end architecture to embed the semantic representation into the visual space. Different from the above works, we choose not to learn the inverse mapping directly but generate synthesized features of unseen classes conditioned on class-level semantic attributes. Recently, some works focused on data generation using generative models, which are similar to our work. For instance, Bucher et al. [5] generated features via GMMN [22]. Xian et al. [43] proposed a framework combining WGAN [3] and a categorization network to generate features. Our framework differs from them by exploiting two generative models (*i.e.* C-VAE and CGAN) for realistic feature generation. Moreover, we propose both categorization and perceptual losses to generate discriminative features.

In comparison to the conventional ZSL, the generalized zero-shot learning (GZSL) is a more realistic and difficult task, where both seen and unseen classes are available at test time [7, 42]. Despite that the conventional ZSL has gained a lot of attention, few studies [7, 37] concentrated on solving GZSL problems. It is more desirable to design robust ZSL methods that could eliminate the bias towards the seen data for more realistic scenarios.

2.2 Deep Generative Models

Deep generative models [13, 15] have shown the great potential in data generation. There have been a variety of deep generative models [13, 15, 20, 21, 32, 38]. Among these models, Variational Autoencoder (VAE) [15] and Generative Adversarial Network (GAN) [13] play the indispensable roles. VAE models the relationship directly through element-wise reconstruction, while GAN captures the global relationship indirectly [21]. However, VAE has a disadvantage of often generating blurry images as reported in [4] because element-wise distance cannot describe the complex data structure. GAN can obtain more abstract information, but the training process is not stable [36].

Due to the above shortcomings, some recent works attempted to combine these two generative models for better data generation, such as VAE/GAN [21], adversarial autoencoder [25], and CVAE-GAN [4]. Our work is thus motivated by the above approaches; however, we utilize conditioned generative models to synthesize features instead of images as the quality of generated images are too low to achieve satisfactory performance in ZSL problems [43]. Specifically, our model is conditioned on semantic attributes instead of category-level labels, so that more delicate description can be used for feature generation. Moreover, we add a categorization network to ensure that the generated features are helpful for the following classification task. We also take advantage of the intermediate outputs of the networks for perceptual reconstruction to form a richer semantic similarity metric for feature generation.

3 Approach

This work aims to synthesize high-quality features for unseen classes by establishing a joint generative model, based on which conventional ZSL can be

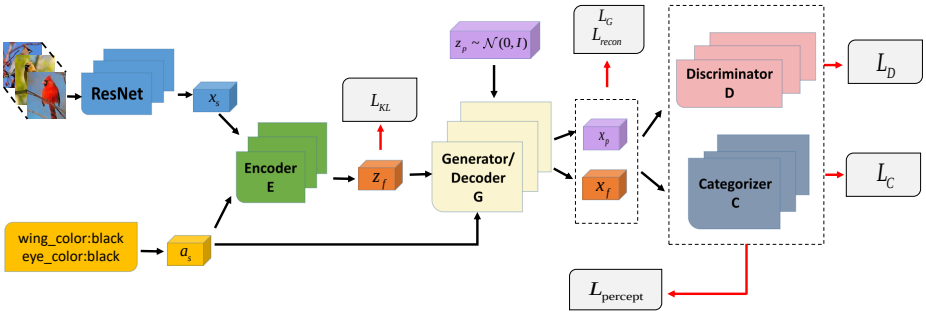


Fig. 2. The illustration of our generative framework. Particularly, our framework consists of four networks: the Encoder E, the Generator G, the Discriminator D, and the Categorizer C. Given real features and the corresponding semantic attributes as the input, our framework will synthesize high-quality features after generative learning.

transformed into supervised learning. Specifically, our proposed model generates semantically expressive features for unseen classes conditioned on the class-level semantic attributes. Subsequently, we train classifiers based on the generated features of unseen classes w.r.t. conventional ZSL settings and on both the generated features of unseen classes and real features of seen classes w.r.t. GZSL settings. As a result, the domain shift between seen and unseen classes will be mitigated significantly as classifiers are learned on both seen and unseen features.

In the following, we will first introduce the problem settings for ZSL and GZSL, and then present our joint generative model in detail. Finally, how to perform zero-shot recognition in a supervised manner is elaborated.

3.1 Problem Settings

In zero-shot learning, the training set S consists of image features, attributes, and class labels of seen classes, *i.e.* $S = \{(x_s, a_s, y_s)|x_s \in X, a_s \in A, y_s \in Y_s\}$. $x_s \in R^{d_x}$ denotes the features of seen data, where d_x denotes the feature dimension. $Y_s = \{y_s^1, ..., y_s^{C_s}\}$ represents the labels of C_s seen classes. $a_s \in \mathbb{R}^{d_a}$ denotes the class-level attributes of seen classes, where d_a indicates the dimension of semantic attributes. In terms of unseen classes, no features are available during training and we can only employ some class-level information, *e.g.* semantic attributes in our case. Specifically, the unseen set is denoted by $U = \{(a_u, y_u)|a_u \in A, y_u \in Y_u\}$, where $Y_u = \{y_u^1, ..., y_u^{C_u}\}$ represents the labels of C_u unseen classes and $a_u \in \mathbb{R}^{d_a}$ denotes the class-level attributes of unseen classes.

It should be noted that the seen and unseen classes are disjoint, namely $Y_s \cap Y_u = \emptyset$. Given S and U, the conventional zero-shot learning task is to learn a classifier $f_{ZSL} : X \rightarrow Y_u$, and the generalized zero-shot learning aims to learn a universal classifier $f_{GZSL} : X \rightarrow Y_s \cup Y_u$, which is a more challenging task.

3.2 Joint Generative Model

In this subsection, we will introduce our proposed framework in detail. As shown in Figure 2, our framework consists of four networks: **1)** the encoder network E, **2)** the decoder/generator network G, **3)** the discriminator network D, and **4)** the categorizer network C. As our framework combines CVAE and CGAN, the decoder in CVAE is identical to the generator in CGAN. Unless otherwise specified, we use the generator G to denote this network branch.

The combination of CVAE and CGAN provides well-designed guidance for feature generation. In the following, we will first introduce the network structures of VAE conditioned on semantic attributes and GAN conditioned on semantic attributes and category labels, respectively. An additional categorization network will also be introduced along with the conditional GAN. Subsequently, we will present our perceptual reconstruction loss and the overall objective for training in detail. Finally, we will introduce the procedure for zero-shot recognition at test time.

VAE Conditioned on Semantic Attributes VAE consists of an encoder network and a generator network. In our architecture, VAE is conditioned on class-level semantic attributes. In other words, attributes act as a part of the input to both encoder and generator for the purpose of providing class-level semantic information for feature generation.

As for the encoder network E with parameters θ_E, we aim to encode the real features x_s into a latent representation

$$z_f \sim p_E(z|x_s, a_s), \tag{1}$$

where $x_s \sim p(x)$ and $a_s \sim p(a)$, and $p(x)$ and $p(a)$ denote the prior distributions of real features and semantic attributes, respectively. The encoder learns the inherent structure of features and then imposes this prior over the distribution $p(z)$, which is usually $z_p \sim \mathcal{N}(0, I)$. The generator G with parameters θ_G decodes the latent representation into the feature space to generate synthesized features

$$x_f \sim p_G(x|z_f, a_s). \tag{2}$$

The overall loss function of CVAE is a combination of the reconstruction loss and the Kullback-Leibler divergence loss:

$$L_{CVAE}(\theta_E, \theta_G) = L_{KL} + L_{recon}, \tag{3}$$

where

$$L_{KL}(\theta_E, \theta_G) = KL(p_E(z|x_s, a_s)||p(z)), \tag{4}$$

$$L_{recon}(\theta_E, \theta_G) = -E[\log(p_G(x|z_f, a_s))]. \tag{5}$$

By minimizing Eq. (3), we can reduce the reconstruction error and the difference between the distribution of latent representation and the prior distribution. As a consequence, the encoder is capable of capturing the inherent structure of data and the generator will generate features with similar structures as the real ones.

GAN Conditioned on Attributes and Categories In the conventional generative adversarial network, the generator and the discriminator try to make a balance in a two-player minimax competition. In our framework, the generator is conditioned on the semantic attributes. In addition, the category-wise information (*i.e.* labels), exploited by a categorizer, works as another clue to help the generator obtain discriminative features. We define the discriminator with parameters θ_D and the categorizer with parameters θ_C. Concretely, the generator tries to minimize the following loss:

$$\begin{aligned} L_G(\theta_G, \theta_D, \theta_C) = &- E[\log(p_D(G(z_p, a_s)))] - E[\log(p_D(G(z_f, a_s)))] \\ &- E[\log(p_C(y_s|x_p)] - E[\log(p_C(y_s|x_f)], \end{aligned} \tag{6}$$

where

$$x_p = G(z_p, a_s) \sim p_G(x|z_p, a_s), x_f = G(z_f, a_s) \sim p_G(x|z_f, a_s).$$

In the meantime, the discriminator tries to minimize

$$L_D(\theta_G, \theta_D, \theta_C) = -E[\log(p_D(x_s))] - E[\log(1 - p_D(x_f))] - E[\log(1 - p_D(x_p))] \tag{7}$$

Given z_p and z_f along with the semantic attributes as the input, the generator aims to synthesize features that are similar to the real features and belong to the same class as the real ones at the same time. The discriminator tries to distinguish real features from synthesized ones. After iterative training, the network will generate high-quality features with the guidance from semantic attributes as well as from category-wise information.

As mentioned above, the categorizer helps to promote the discriminability of the generated features, which has the similar spirit with the classification network in [43]. However, we find that this additional regularization is not enough for the subsequent classification task. To this end, we make the categorizer as the other 'discriminator', which plays a minimax competition with the generator in the category level. Concretely, the real features x_s, and synthesized features x_f and x_p, are fed into the categorizer, which tries to minimize the softmax based categorization loss:

$$L_C(\theta_C) = -E[\log(p_C(y_s|x_s)] - E[\log(p_C(y_f|x_p)] - E[\log(p_C(y_f|x_f)], \tag{8}$$

where y_f denotes the label of the 'fake' class that is disjoint from the seen and unseen classes. In this way, the categorizer not only needs to classify the real features into the right classes but also regards the synthesized features as another 'fake' class. Through the competition, the generator is encouraged to generate features from the same classes as the real features.

Perceptual Reconstruction In addition to the superior characteristics of C-VAE and CGAN, we try to find a richer similarity metric to achieve more delicate generation results. As we mentioned above, element/pixel-wise information and holistic structures can be preserved by using VAE and GAN respectively, yet the

semantic information may not be enough. Thus, we incorporate a perceptual reconstruction loss into our framework. The perceptual loss has been explored in the field of image style transfer and super-resolution [14], and could encourage the generated features to be semantically similar to real ones.

Specifically, we take advantage of the intermediate output of the discriminator and categorizer for perceptual reconstruction:

$$L_{percept}(\theta_D, \theta_C) = \|f_D(x_s) - f_D(x_f)\|_2^2 + \|f_C(x_s) - f_C(x_f)\|_2^2, \tag{9}$$

where f_D and f_C are the outputs of the last hidden layers of the discriminator and categorizer, respectively.

Overall Objective The ultimate goal of our framework is to minimize the following overall loss function:

$$L = L_{KL} + L_{recon} + L_D + L_C + \alpha L_G + \beta L_{percept}. \tag{10}$$

In particualr, we alternatively optimize every network branch in our framework as follows:

$$Encoder(\theta_E) \leftarrow L_{KL} + L_{recon} + \beta L_{percept}; \tag{11}$$

$$Generator(\theta_G) \leftarrow L_{recon} + \alpha L_G + \beta L_{percept}; \tag{12}$$

$$Discriminator(\theta_D) \leftarrow L_D; \tag{13}$$

$$Categorizer(\theta_C) \leftarrow L_C. \tag{14}$$

L_{KL} only appears in Eq. (11) because it is only related to the encoder. Similarly, L_C and L_D are the objectives of the categorizer and discriminator respectively. The generator is shared between the CVAE and CGAN so its loss can be divided into two parts: *i.e.* L_{recon} and $L_{percept}$ form the loss w.r.t. CVAE and L_G is the loss w.r.t. CGAN. All the objectives are complementary to each other, while the joint training process could result in superior performance.

3.3 Zero-Shot Recognition

After finishing the training process, the synthesized features of unseen classes can be obtained through our generator network. In particular, given an arbitrary latent representation drawn from the Gaussian distribution $z_t \sim \mathcal{N}(0, I)$ and the semantic attributes a_u of the corresponding unseen class as the input, the generator will output the synthesized features as follows:

$$x_{gen} = G(z_t, a_u) \sim p_G(x|z_t, a_u). \tag{15}$$

Based on the generated features, zero-shot recognition can be transformed into the conventional supervised learning problem. As we previously mentioned, there exist two settings for zero-shot recognition, *i.e.* the conventional ZSL and the more challenging GZSL. In conventional ZSL settings, we train the softmax classifier based on x_{gen} and then test on the real features of unseen classes,

Algorithm 1 The training process of our proposed framework

Input:

Training features of seen classes: x_s; semantic attributes of seen classes: a_s; initial parameters of Encoder E, Generator G, Discriminator D, and Categorizer C: θ_E, θ_G, θ_D, and θ_C; total training epoch: T.

Output:

The learned parameters of each network: θ_E, θ_G, θ_D, and θ_C.

while epoch < T **do**
  Sample a batch of real features $\{x_s, a_s\} \subseteq S$;
  The Encoder E maps the real features into a latent representation: z_f;
  Compute KL loss using Eq. (4);
  Get synthesized features x_f through the Generator G;
  Compute reconstruction loss using Eq. (5);
  Sample z_p from the Gaussian distribution: $z_p \sim \mathcal{N}(0, I)$;
  Get synthesized features x_p through the Generator G;
  Compute the generator loss using Eq. (6);
  Compute the discriminator loss using Eq. (7);
  Classify x_s and synthesized features x_f, x_p through the Categorizer C;
  Compute the categorization loss using Eq. (8);
  Compute the perceptual reconstruction loss using Eq. (9);
  Optimizing the parameters of each network using Eq. (11) - (14), respectively;
end while

i.e. x_u. As for GZSL settings, the original data of seen classes x_s will be divided into two parts, *i.e.* x_s^{tr} for training and x_s^{ts} for test. During training, we employ x_{gen} together with x_s^{tr} as the training samples to learn the softmax classifier. At test time, we evaluate on x_u and x_s^{ts} to obtain the final recognition accuracy.

4 Experimental Results

In this section, we evaluate the proposed method on five ZSL benchmark datasets. First, we make a brief introduction of the datasets, implementation details of our framework and evaluation protocols. In order to show the effectiveness of our framework, we then present our experimental results on both conventional ZSL and GZSL tasks by comparing with several state-of-the-art ZSL methods and baseline methods. Finally, we show the high quality of the generated features quantitatively and qualitatively.

4.1 Datasets

Five classic datasets for ZSL are adopted in our experiments, *i.e.* AWA1 [18], AWA2 [42], CUB [40], SUN [29], and aPY [8]. AWA1 [18] is the original Animals with Attributes dataset, which has 30475 images in 50 classes, and each class is annotated with 85 attributes. However, the images of AWA1 are not publicly available. The AWA2 [42] dataset, containing 37322 images, is a good replacement for AWA1. These two datasets share the same classes and class-level

Table 1. Statistics of datasets in term of number of images, attributes, and seen/unseen classes, and the training/test split.

Dataset	Image	Attribute	Seen/Unseen	Training		Test	
				Seen	Unseen	Seen	Unseen
CUB [40]	11788	312	150/50	7057	0	1764	2967
AWA1 [18]	30475	85	40/10	19832	0	4958	5685
AWA2 [42]	37322	85	40/10	23527	0	5882	7913
SUN [29]	14340	102	645/72	10320	0	2580	1440
aPY [8]	15539	64	20/12	5932	0	1483	7924

attributes. Caltech-UCSD Birds 200-2011 (CUB) [40] is a fine-grained dataset with 11788 images of birds of 200 different types annotated with 312 attributes. SUN [29] is also a fine-grained dataset that contains 14340 images from 717 types of scenes annotated with 102 attributes. Attribute Pascal and Yahoo (aPY) [8] is a small-scale dataset with 15339 images from 32 classes annotated with 64 attributes. The details of the five datasets are summarized in Table 1.

As for image features, we employ the ResNet features proposed in [42]. Regarding class embeddings, we use the class-level continuous attributes for all datasets because using continuous attributes could achieve better performance than binary ones, as pointed out in [1]. As for data splits, in early standard splits [18] for each dataset, some of the test classes are among the 1K classes of ImageNet, which are used to pre-train the ResNet. This will lead to biased results. Therefore, we follow the recently proposed split in [42] to avoid this. The detailed seen/unseen splits are also shown in Table 1.

4.2 Implementation Details and Parameter Settings

In our framework, all the networks are Multi-Layer Perceptrons (MLP) with LeakyReLU activations [44]. The encoder, generator, and discriminator consist of a single hidden layer with 1000 units, and the categorizer contains a single hidden layer with 1024 units. As each dataset has different attribute annotations, we set the dimension d_z of z_f and z_p according to the number of class-level attributes respectively. Specifically, we set $d_z = 256$ for AWA1, AWA2, SUN, and aPY, and $d_z = 512$ for CUB as CUB dataset has much more attributes.

For network training, we first pre-train the categorizer branch using the seen data for fast convergence. In terms of the parameters, we empirically set $\alpha = 0.01$, and $\beta = 0.1$ across all the datasets. The number of the generated features are chosen to make a trade-off between the computational efficiency and classification accuracy. Specifically, in the conventional ZSL task, we set the number of generated features as eight times the number of ground-truth unseen features on CUB, SUN and aPY, and twice on AWA1 and AWA2. As for GZSL, we set the number of generated features as eight times the number of ground-truth unseen features on SUN and aPY, and six times on CUB, AWA1, and AWA2.

Table 2. Comparison results with the state-of-the-art methods in terms of both ZSL and GZSL settings. T1 = top-1 accuracy, u = top-1 accuracy on unseen data, s = top-1 accuracy on seen data, and H = harmonic mean. We report top-1 accuracies in %.

Method	Zero-Shot Learning					Generalized Zero-Shot Learning														
	CUB	AWA1	AWA2	SUN	aPY	CUB			AWA1			AWA2			SUN			aPY		
	T1	T1	T1	T1	T1	u	s	H	u	s	H	u	s	H	u	s	H	u	s	H
DAP[18]	40.0	44.1	46.1	39.9	33.8	1.7	67.9	3.3	0.0	**88.7**	0.0	0.0	84.7	0.0	4.2	25.1	7.2	4.8	78.3	9.0
IAP[18]	24.0	35.9	35.9	19.4	36.6	0.2	**72.8**	0.4	2.1	78.2	4.1	0.9	87.6	1.8	1.0	37.8	1.8	5.7	65.6	10.4
CONSE[27]	34.3	45.6	44.5	38.8	26.9	1.6	72.2	3.1	0.4	88.6	0.8	0.5	**90.6**	1.0	6.8	39.9	11.6	0.0	**91.2**	0.0
CMT[37]	34.6	39.5	37.9	39.9	28.0	7.2	49.8	12.6	0.9	87.6	1.8	0.5	90.0	1.0	8.1	21.8	11.8	1.4	85.2	2.8
SSE[47]	43.9	60.1	61.0	51.5	34.0	8.5	46.9	14.4	7.0	80.5	12.9	8.1	82.5	14.8	2.1	36.4	4.0	0.2	78.9	0.4
LATEM[41]	49.3	55.1	55.8	55.3	35.2	15.2	57.3	24.0	7.3	71.7	13.3	11.5	77.3	20.0	14.7	28.8	19.5	0.1	73.0	0.2
ALE[1]	54.9	59.9	62.5	58.1	39.7	23.7	62.8	34.4	16.8	76.1	27.5	14.0	81.8	23.9	21.8	33.1	26.3	4.6	73.7	8.7
DEVISE[10]	52.0	54.2	59.7	56.5	**39.8**	23.8	53.0	32.8	13.4	68.7	22.4	17.1	74.7	27.8	16.9	27.4	20.9	4.9	76.9	9.2
SJE[2]	53.9	65.6	61.9	53.7	32.9	23.5	59.2	33.6	11.3	74.6	19.6	8.0	73.9	14.4	14.7	30.5	19.8	3.7	55.7	6.9
ESZSL[35]	53.9	58.2	58.6	54.5	38.3	12.6	63.8	21.0	6.6	75.6	12.1	5.9	77.8	11.0	11.0	27.9	15.8	2.4	70.1	4.6
SYNC[6]	55.6	54.0	46.6	56.3	23.9	11.5	70.9	19.8	8.9	87.3	16.2	10.0	90.5	18.0	7.9	**43.3**	13.4	7.4	66.3	13.3
SAE[16]	33.3	53.0	54.1	40.3	8.3	7.8	54.0	13.6	1.8	77.1	3.5	1.1	82.2	2.2	8.8	18.0	11.8	0.4	80.9	0.9
f-CLSWGAN[43]	57.3	68.2	-	60.8	-	**43.7**	57.7	**49.7**	57.9	61.4	59.6	-	-	-	42.6	36.6	**39.4**	-	-	-
SE-GZSL[39]	**59.6**	69.5	69.2	**63.4**	-	41.5	53.3	46.7	56.3	67.8	61.5	**58.3**	68.1	62.8	40.9	30.5	34.9	-	-	-
Proposed	54.9	**69.9**	**69.5**	59.0	36.3	42.7	45.6	44.1	**62.7**	60.6	**61.6**	56.2	71.7	**63.0**	**44.4**	30.9	36.5	**31.1**	43.3	**36.2**

4.3 Evaluation Protocol

As mentioned above, in conventional ZSL settings, we aim to classify the unseen features x_u into the corresponding unseen classes Y_u. In GZSL settings, the class space is $Y_s \cup Y_u$ and we need to assign class labels to both unseen features and some of the seen features. Here we follow the unified evaluation protocol in [42].

In the conventional ZSL setting, we compute the average top-1 accuracy for each class and then average the per-class top-1 accuracy to mitigate the imbalance among the classes. The evaluation metric is defined as follows:

$$acc = \frac{1}{\|C\|} \sum_{c=1}^{\|C\|} \frac{n_{cp}}{n_c}, \tag{16}$$

where $\|C\|$ denotes the number of classes, n_c denotes the number of data in each class, and n_{cp} is the number of correct predictions in each class. Regarding GZSL, we use the harmonic mean, which can be computed as follows:

$$H = \frac{2 * s * u}{s + u} \tag{17}$$

where s and u represent the average per-class top-1 accuracies of seen classes and unseen classes respectively. A higher harmonic mean indicates the high accuracies on both seen and unseen classes.

4.4 Comparison with the State-of-the-Art Methods

Table 2 shows the conventional ZSL results of our framework and the state-of-the-art methods. In this setting, the search space is restricted to unseen classes at test time. From the table, we can observe that our method achieves better

Table 3. Comparison results with the baseline models in terms of both ZSL and GZSL settings. T1 = top-1 accuracy, u = top-1 accuracy on unseen data, s = top-1 accuracy on seen data, and H = harmonic mean. We report top-1 accuracies in %.

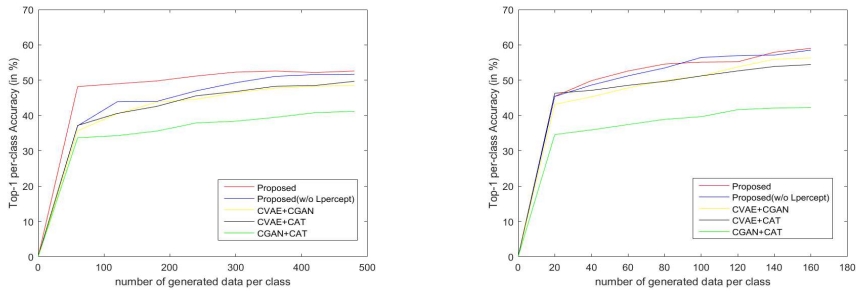
Method	Zero-Shot Learning					Generalized Zero-Shot Learning														
	CUB	AWA1	AWA2	SUN	aPY	CUB			AWA1			AWA2			SUN			aPY		
	T1	T1	T1	T1	T1	u	s	H	u	s	H	u	s	H	u	s	H	u	s	H
CVAE+CAT	48.7	65.0	65.2	54.4	32.0	33.3	**54.0**	41.2	39.3	75.9	55.7	34.7	**83.3**	49.0	45.1	25.5	32.6	20.4	48.5	28.7
CGAN+CAT	41.2	59.6	56.6	42.3	17.3	0.0	41.8	0.0	10.8	**76.9**	19.0	12.0	82.8	20.9	0.0	**41.1**	0.0	10.3	89.6	18.5
CVAE+CGAN	48.6	65.4	59.8	56.3	33.7	38.4	42.6	40.4	46.5	70.5	56.0	41.8	77.0	54.1	38.8	29.2	33.3	22.0	**96.7**	33.4
Proposed (w/o $L_{percept}$)	51.1	68.4	66.2	58.5	34.9	40.5	7.8	43.9	50.5	67.8	57.9	51.7	74.8	61.1	**49.0**	26.0	34.0	30.8	37.5	33.8
Proposed	**54.9**	**69.9**	**69.5**	**59.0**	**36.3**	**42.7**	45.6	**44.1**	**62.7**	60.6	**61.6**	**56.2**	71.7	**63.0**	44.4	30.9	**36.5**	**31.1**	43.3	**36.2**

zero-shot recognition accuracies than the traditional ZSL methods. The overall improvement is especially obvious on AWA1 and AWA2, with 6.6% and 11.2% higher accuracies than the second best ones in traditional ZSL methods, respectively. Compared with the generative models in ZSL tasks, our method have better performance on AWA1 and AWA2 datasets, with 0.6% higher accuracies on both of the datasets. Concerning CUB, SUN and aPY datasets, our method performs competitively with the best ones, *i.e.* SE-GZSL [39] and DEVISE [10] methods, respectively. The results clearly demonstrate that our framework is capable of generating useful and expressive features of unseen classes, which are beneficial for ZSL tasks.

In terms of the GZSL task, as illustrated in Table 2, our framework shows the superiority over the traditional ZSL methods on all the five datasets. For example, significant improvements w.r.t. harmonic mean are observed, with 172.2%, 123.7%, 122.2%, 32.7%, and 28.2% higher than the second best ones on aPY, AWA2, AWA1, SUN, and CUB, respectively. It is noteworthy that most traditional ZSL methods achieve high accuracies on seen classes but much worse performance on unseen classes, which indicates that those methods have strong biases towards seen classes. Our model can mitigate the bias to a large extent as shown in Table 2. Compared with the ZSL methods based on the generative models, our model shows the superiorities on AWA1 and AWA2 datasets. Moreover, our model has the highest accuracy for unseen classes on AWA1, SUN and aPY datasets, indicating that our model has the capability of balancing the accuracy between seen and unseen classes. Therefore, our generative framework is very useful and competitive in this realistic and challenging task.

4.5 Comparison with the Baseline Models

As our framework contains several networks together with the perceptual reconstruction, we compare the proposed framework with four baseline models by omitting each of them, in order to verify the importance of each branch. For example, as shown in Table 3, 'CVAE+CAT' indicates the framework only containing the CVAE and categorizer, and 'Proposed (w/o $L_{percept}$)' denotes the whole network without the perceptual reconstruction.



(a) CUB (b) SUN

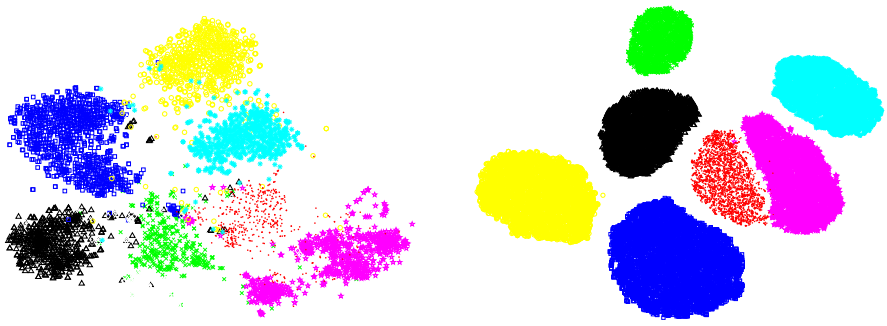
Fig. 3. Top-1 accuracy with different numbers of generated features on the CUB and SUN datasets.

The results w.r.t. conventional ZSL settings are shown in Table 3. From the results of 'CVAE+CAT' and 'CGAN+CAT', we can conclude that integrating the CVAE and CGAN are beneficial for the ZSL task, and the improvement is more significant by incorporating the CVAE. This shows that element-wise reconstruction is essential for our task. The results of 'CVAE+CGAN' also demonstrate the necessity of the categorizer branch. For example, the accuracy is improved by 7.7% on aPY by adding the categorizer. Finally, we can see that our framework with perceptual reconstruction outperforms the one without $L_{percept}$.

As for GZSL, compared with the above baselines, the proposed model has higher accuracies on unseen classes and higher harmonic mean since it can balance the seen and unseen classes. Overall, 'CGAN+CAT' achieves the worst performance probably because CGAN captures the holistic data structure, which is not enough for feature generation. After combining the CVAE, the performance is enhanced significantly. All the above results in ZSL and GZSL settings clearly demonstrate the indispensability of each part in our whole framework.

4.6 Analysis of the Generated Features

In this section, we present some further analyses of the synthesized features of unseen classes. Figure 3 shows the classification accuracies for the unseen classes with the increasing numbers of the generated features. In general, the accuracy increases when generating more unseen features. We also observe that the satisfactory accuracies can be achieved when the numbers of the generated features are relatively small, which indicates that our model can generate high-quality features for the classification task. The generated features can be used as the excellent replacement of the missing unseen features. Taking some unseen classes on the AWA1 dataset as an example, we can see from Figure 4 that the generated feature distribution is even more discriminative compared with the



(a) Real features (b) Generated features

Fig. 4. t-SNE visualization of the real/generated features of some unseen classes on the AWA1 dataset.

real feature distribution. This further indicates that our generative model can synthesize high-quality features that are beneficial for the classification task.

5 Conclusion

In this work, we proposed an effective joint generative framework for feature generation in the context of zero-shot learning. Specifically, our model combined two popular generative models, *i.e.* VAE and GAN, to capture the element-wise and holistic data structures at the same time. We took advantage of the class-level semantic attributes as the conditional information. An additional categorization network worked as the guidance for generating discriminative features. Importantly, we incorporated the perceptual reconstruction into the framework to preserve semantic similarities. We showed the superiority of the proposed generative framework by conducting experiments on five standard datasets in terms of the conventional ZSL task as well as the more challenging GZSL task. The extensive experimental results indicated that our model could generate high-quality features to mitigate the domain gap in ZSL due to the lack of unseen data.

Acknowledgements

This work was supported in part by the NSFC under Grant 61872286, u1531141, 61732008, 61772407 and 61701391, the National Key R&D Program of China under Grant 2017YFF0107700, the National Science Foundation of Shaanxi Province under Grant 2018JM6092, and Guangdong Provincial Science and Technology Plan Project under Grant 2017A010101006 and 2016A010101005.

References

1. Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C.: Label-embedding for attribute-based classification. In: CVPR (2013)
2. Akata, Z., Reed, S., Walter, D., Lee, H., Schiele, B.: Evaluation of output embeddings for fine-grained image classification. In: CVPR (2015)
3. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. In: ICML (2017)
4. Bao, J., Chen, D., Wen, F., Li, H., Hua, G.: Cvae-gan: fine-grained image generation through asymmetric training. In: ICCV (2017)
5. Bucher, M., Herbin, S., Jurie, F.: Generating visual representations for zero-shot classification. In: ICCV Workshop (2017)
6. Changpinyo, S., Chao, W.L., Gong, B., Sha, F.: Synthesized classifiers for zero-shot learning. In: CVPR (2016)
7. Chao, W.L., Changpinyo, S., Gong, B., Sha, F.: An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In: ECCV (2016)
8. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: CVPR (2009)
9. Ferrari, V., Zisserman, A.: Learning visual attributes. In: NIPS (2008)
10. Frome, A., Corrado, G.S., Shlens, J., Bengio, S., Dean, J., Mikolov, T., et al.: Devise: A deep visual-semantic embedding model. In: NIPS (2013)
11. Fu, Y., Hospedales, T.M., Xiang, T., Gong, S.: Attribute learning for understanding unstructured social activity. In: ECCV (2012)
12. Fu, Y., Hospedales, T.M., Xiang, T., Gong, S.: Transductive multi-view zero-shot learning. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(11), 2332–2345 (2015)
13. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS (2014)
14. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: ECCV (2016)
15. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2014)
16. Kodirov, E., Xiang, T., Gong, S.: Semantic autoencoder for zero-shot learning. In: CVPR (2017)
17. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: CVPR (2009)
18. Lampert, C.H., Nickisch, H., Harmeling, S.: Attribute-based classification for zero-shot visual object categorization. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(3), 453–465 (2014)
19. Larochelle, H., Erhan, D., Bengio, Y.: Zero-data learning of new tasks. In: AAAI (2008)
20. Larochelle, H., Murray, I.: The neural autoregressive distribution estimator. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (2011)
21. Larsen, A.B.L., Sønderby, S.K., Larochelle, H., Winther, O.: Autoencoding beyond pixels using a learned similarity metric. In: ICML (2016)
22. Li, Y., Swersky, K., Zemel, R.: Generative moment matching networks. In: ICML (2015)
23. Long, Y., Liu, L., Shao, L.: Towards fine-grained open zero-shot learning: Inferring unseen visual features from attributes. In: WACV (2017)
24. Long, Y., Liu, L., Shao, L., Shen, F., Ding, G., Han, J.: From zero-shot learning to conventional supervised classification: Unseen visual data synthesis. In: CVPR (2017)

25. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B.: Adversarial autoencoders. arXiv preprint arXiv:1511.05644 (2015)
26. Mirza, M., Osindero, S.: Conditional generative adversarial nets. Computer Science pp. 2672–2680 (2014)
27. Norouzi, M., Mikolov, T., Bengio, S., Singer, Y., Shlens, J., Frome, A., Corrado, G.S., Dean, J.: Zero-shot learning by convex combination of semantic embeddings. In: ICLR (2014)
28. Palatucci, M., Pomerleau, D., Hinton, G.E., Mitchell, T.M.: Zero-shot learning with semantic output codes. In: NIPS (2009)
29. Patterson, G., Hays, J.: Sun attribute database: Discovering, annotating, and recognizing scene attributes. In: CVPR (2012)
30. Qin, J., Liu, L., Shao, L., Shen, F., Ni, B., Chen, J., Wang, Y.: Zero-shot action recognition with error-correcting output codes. In: CVPR (2017)
31. Qin, J., Wang, Y., Liu, L., Chen, J., Shao, L.: Beyond semantic attributes: Discrete latent attributes learning for zero-shot recognition. IEEE Signal Process. Lett. **23**(11), 1667–1671 (2016)
32. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. Computer Science (2015)
33. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. In: ICML (2016)
34. Rohrbach, M., Stark, M., Schiele, B.: Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In: CVPR (2011)
35. Romera-Paredes, B., Torr, P.: An embarrassingly simple approach to zero-shot learning. In: ICML (2015)
36. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: NIPS (2016)
37. Socher, R., Ganjoo, M., Manning, C.D., Ng, A.: Zero-shot learning through cross-modal transfer. In: NIPS (2013)
38. Sohn, K., Lee, H., Yan, X.: Learning structured output representation using deep conditional generative models. In: NIPS (2015)
39. Verma, V.K., Arora, G., Mishra: Generalized zero-shot learning via synthesized examples. In: CVPR (2018)
40. Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., Perona, P.: Caltech-ucsd birds 200. California Institute of Technology (2010)
41. Xian, Y., Akata, Z., Sharma, G., Nguyen, Q., Hein, M., Schiele, B.: Latent embeddings for zero-shot classification. In: CVPR (2016)
42. Xian, Y., Lampert, C.H., Schiele, B., Akata, Z.: Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. In: CVPR (2017)
43. Xian, Y., Lorenz, T., Schiele, B., Akata, Z.: Feature generating networks for zero-shot learning. In: CVPR (2018)
44. Xu, B., Wang, N., Chen, T., Li, M.: Empirical evaluation of rectified activations in convolutional network. Computer Science (2015)
45. Yu, X., Aloimonos, Y.: Attribute-based transfer learning for object categorization with zero/one training example. In: ECCV (2010)
46. Zhang, L., Xiang, T., Gong, S., et al.: Learning a deep embedding model for zero-shot learning. In: CVPR (2017)
47. Zhang, Z., Saligrama, V.: Zero-shot learning via semantic similarity embedding. In: ICCV (2015)
48. Zhang, Z., Saligrama, V.: Zero-shot recognition via structured prediction. In: ECCV (2016)