Self-supervised Domain-aware Generative Network for Generalized Zero-shot Learning

Jiamin Wu[1], Tianzhu Zhang[1,*], Zheng-Jun Zha[1], Jiebo Luo[2], Yongdong Zhang[1], Feng Wu[1]
[1] University of Science and Technology of China
[2] University of Rochester
jiaminwu@mail.ustc.edu.cn, {tzzhang, zhazj, zhyd73, fengwu}@ustc.edu.cn
jluo@cs.rochester.edu

Abstract

Generalized Zero-Shot Learning (GZSL) aims at recognizing both seen and unseen classes by constructing correspondence between visual and semantic embedding. However, existing methods have severely suffered from the strong bias problem, where unseen instances in target domain tend to be recognized as seen classes in source domain. To address this issue, we propose an end-to-end Self-supervised Domain-aware Generative Network (SDGN) by integrating self-supervised learning into feature generating model for unbiased GZSL. The proposed SDGN model enjoys several merits. First, we design a cross-domain feature generating module to synthesize samples with high fidelity based on class embeddings, which involves a novel target domain discriminator to preserve the domain consistency. Second, we propose a self-supervised learning module to investigate inter-domain relationships, where a set of anchors are introduced as a bridge between seen and unseen categories. In the shared space, we pull the distribution of target domain away from source domain, and obtain domain-aware features with high discriminative power for both seen and unseen classes. To our best knowledge, this is the first work to introduce self-supervised learning into GZSL as a learning guidance. Extensive experimental results on five standard benchmarks demonstrate that our model performs favorably against state-of-the-art GZSL methods.

1. Introduction

Zero-Shot Learning (ZSL) [1, 8, 29, 48, 34] aims to recognize images of unseen classes without any labeled samples available. This requires the ZSL approaches to comprehend images beyond the visual level, which is generally achieved by semantic bridging and knowledge transferring between seen and unseen classes, with the aid of semantic information like attributes [8], word embeddings [23] and text descriptions [28].

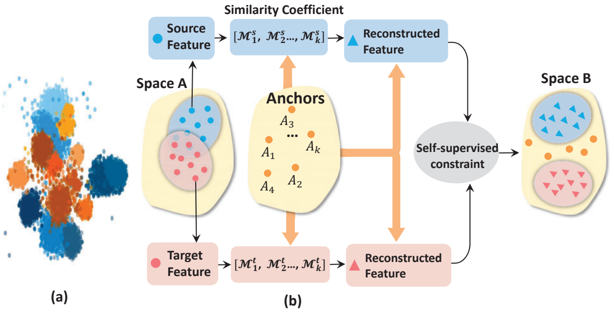


Figure 1. (a) The t-SNE visualization of features on AwA1. Orange and blue color denote source and target domain. (b) Our motivation. Space A is the visual space generated by common models, while Space B in SDGN is augmented by a self-supervised constraint, which pushes target data away from source data.

Earlier efforts on ZSL [4, 37, 41, 29] are focused on the cross-modal mapping between visual feature and semantic embedding to seek a shared space. Recently, a plethora of methods [43, 9, 13, 14] propose to utilize generative models to synthesize visual samples with class embeddings, which can be used for training a standard classifier for both seen and unseen classes. Here, the class embedding means that each class is embedded in the space of attribute vectors. GAN-based ZSL methods [45, 27] show stronger performances compared with cross-modal mapping methods, since their classifiers are directly trained with synthesized data. However, in a more practical setting, where both seen and unseen classes will appear at test time, ZSL suffers from the strong bias problem [5], i.e., instances of unseen classes are more likely to be wrongly classified as one of the seen classes. We call this kind of ZSL as Generalized Zero-Shot Learning (GZSL), and denote seen classes as source domain and unseen classes as target domain. Some methods [4, 1] achieve promising results when test stage comes across only seen classes, but degenerate sharply in generalized setting.

[*] Corresponding Author

The main reason counting for bias problem is the ignorance of domain variance. In most of generative GZSL approaches [43, 17, 7, 9], the GANs are merely used to optimize the divergence between the data distribution of seen classes and generated features, and they do not take unseen classes into consideration. As a result, the generator trained on source features is unable to well accommodate the need for unseen classes due to data variations. Therefore, the authenticity and quality of synthetic data cannot be guaranteed. To deal with the above issue, some methods [27, 22] utilize target unlabeled data in a transductive way. However, these methods take separate operations for different domains, e.g., proposing two GANs or two domain classifiers. They fail to consider the relations between source and target domains, and do not differentiate them from each other. Despite that classes in the same domain may be discriminated well, there is no guarantee that classes from different domains can also be differentiated from each other. As shown in Figure 1 (a), seen classes unnecessarily overlap unseen classes in the visual space, which indicates that they will make disturbance when classifying unseen classes. Therefore, without any supervised information on target domain as a learning guidance, it is very challenging to discover the boundary between target and source domains for comprehensive recognition of all classes in the generalized setting.

To deal with the above limitations, an intuitive idea is to exploit a set of anchors as a bridge between seen classes in source domain and unseen classes in target domain to build a joint embedding space, where the target domain is assumed to be separated from source domain. Both target and source samples can be described uniformly in the space spanned by anchors, where we can discover self-supervision for mining relationships between different domains. As shown in Figure 1 (b), based on anchors, for every sample from seen or unseen class, we are able to compare the sample feature with anchors and derive visual similarities as soft multi-label. Then, we define a weighted combination of all anchors by use of multi-label to obtain the reconstructed feature of this sample. Because classes from seen and unseen domains are completely different, given a sample feature and its reconstructed feature from seen or unseen classes, their similarity should be much higher than the similarity of them with any samples from unseen or seen classes, respectively. Therefore, although without unseen class labels, we can exploit the above self-supervised signal as the learning guidance. Meanwhile, the target multi-label derived from anchors can be integrated into the generative network to help preserve the domain label consistency of synthetic features and their reconstructed features.

Motivated by the above discussions, we propose an end-to-end Self-supervised Domain-aware Generative Network (SDGN) to integrate self-supervised learning into the feature generating model with effective exploitation of unlabeled data. Specifically, we learn a generative model that can synthesize discriminative features for any class of interest, purely based on class embeddings. With these generated features, we introduce anchors to reconstruct domain-specific features, which allows for mapping target and source data into a joint embedding space. To utilize the self-supervised signal, we design a cross-domain triplet mining mechanism, where samples from different domains naturally form negative pairs, and samples with their reconstructed features in the same domain form positive pairs. We integrate these pairs into a triplet loss to investigate the inter-domain relationship. Consequently, the target domain is pushed away from source domain which brings domain-level discriminative power for target classes. Furthermore, we take advantage of the multi-label mentioned above to learn a target domain discriminator, whose input is the pair of visual feature and corresponding multi-label. Optimizing this discriminator will guarantee that the domain labels of synthetic samples and their reconstructed features are consistent with real features in target domain. Taking target data into consideration, our method not only effectively strengthens the knowledge transfer flow from seen data to unseen data but also explores the self-supervised characteristics of the source and target domains.

The contributions of our method could be summarized into three-fold: (1) We propose an end-to-end Self-supervised Domain-aware Generative Network by jointly exploiting a feature generating model and a self-supervised learning module for unbiased GZSL. (2)We design an effective cross-domain triplet mining on the basis of a set of anchors that acts as a bridge between seen and unseen classes, to investigate relationships between source and target domains under the guidance of a self-supervised signal. To our best knowledge, this is the first work to introduce a self-supervised learning strategy into GZSL. (3) Extensive experimental results on five challenging benchmarks demonstrate that our method performs favorably against state-of-the-art GZSL models.

2. Related Work

In this section, we introduce several lines of research in generalized zero-shot learning, transductive zero-shot learning and self-supervised learning.

Generalized Zero-Shot Learning. GZSL is expected to recognize seen and unseen classes by exploiting semantic relations. Earlier GZSL works [29, 11, 34, 1, 20] rely on the cross-modal mapping between the visual and semantic modality. Usually, a compatibility score is calculated between the visual and semantic embedding. Relation Net [36] deploys a deep network to learn an adaptive metric for comparing cross-modal relations. However, due to lack of training samples from unseen classes, the rela-

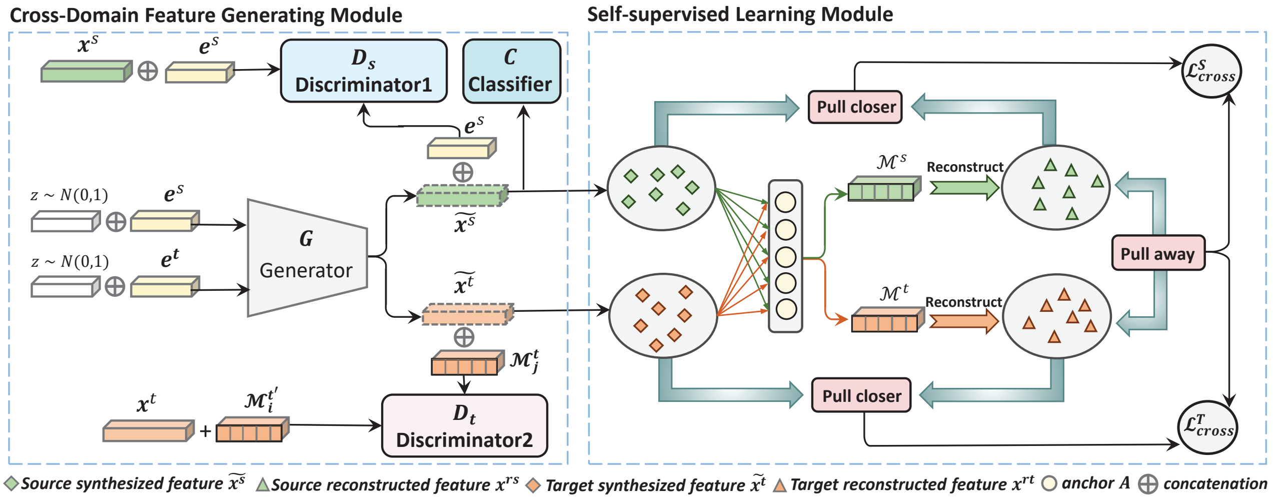


Figure 2. The architecture of SDGN: (1) The cross-domain feature generating module synthesizes features $\tilde{x}^s$ and $\tilde{x}^t$ for seen and unseen classes. It consists of one generator G, and two discriminators D_s and D_t for source and target domain respectively. (2) The self-supervised learning module takes $\tilde{x}^t$ and $\tilde{x}^s$ as input, and compares them with anchors A to derive the soft multi-labels $\mathcal{M}^s$ and $\mathcal{M}^t$ for feature reconstructions. $\tilde{x}$ and its reconstructed feature x^r will be pulled closer while features from different domains will be pulled away.

tionship learned from seen image features and attributes can hardly generalized to unseen classes. Another line of GZSL research [30, 13, 31] follows a feature-generating paradigm. F-CLSWGAN [43] uses a generative model to synthesize visual features. Cycle-CLSWGAN [9] adds a cycle-consistency loss on the feature generation model to make sure the fake features can reconstruct original semantic embeddings. LisGAN [17] utilizes the multi-view meta-representation of each class as guidance for producing more authentic and diverse features. CIZSL [7] imagines new categories that are likely to be unseen by linearly combining source class embeddings to improve the generalization ability of feature synthesis. GAN-based GZSL methods have manifested their advantages over the cross-modal mapping GZSL methods. However, they have a common defect, that is, they do not deal with the appearance variance of different domains, hence perform poorly in the generalized setting.

Transductive Zero-Shot Learning. Transductive ZSL assumes the availability of unlabeled target data [12]. The utilization of unlabeled data is of great diversity. GXE [18] addresses ZSL by generating classifiers from class embeddings, and uses target data to calibrate the classifier generator. QFSL [35] aims at reducing the bias by biasing unseen images to any of the target classes. Some generative models attempt to model the data distribution for unseen categories. SABR [27] trains two GANs to generate features for source and target domains. F-VAEGAN-D2 [45] uses an unconditional discriminator to learn the distribution of unlabeled data. However, none of these methods consider the relation between different domains. They take separate operations for unseen data, but do not construct domain-discriminative representations in a unified framework [19]. Our approach uses the self-supervision lurking in the data structure of different domains to conduct cross-domain mining.

Self-supervised Learning. Self-supervised learning has been widely studied for image and video representation learning. Usually, self-supervised learning extracts supervisory signals from data structure. The signal comes in the forms of temporal order [10, 40], image colorization [16, 49], view points consistency [32] and image completion [25]. SSIAM [33] introduces positive and negative pairs, which are obtained by sorting distances on a frame set, to learn face representation for video face clustering. However, we derive the triplet by inherent relations of source and target domains without metric-based ranking.

3. Our Approach

3.1. Notation

In GZSL, suppose we have the source dataset, i.e., seen classes, defined as $\mathcal{S} = \{(x^s_i, y^s_i)\}, i \in [1, N_s]$, where $x^s_i \in \mathcal{X}_\mathcal{S}$ is the i-th instance of source domain, and $y^s_i \in \mathcal{Y}_\mathcal{S}$ is the seen class label. The target dataset is defined as $\mathcal{T} = \{(x^t_j, y^t_j)\}, j \in [1, N_t]$, where $x^t_j \in \mathcal{X}_\mathcal{T}$ and $y^t_j \in \mathcal{Y}_\mathcal{T}$ denotes the j-th unseen instance and the corresponding label. Here, $\mathcal{Y}_\mathcal{S} \cap \mathcal{Y}_\mathcal{T} = \emptyset$. Class embeddings are denoted as $\mathcal{E} = \{\mathbf{e}_k\}^{\mathcal{C}}_{k=1}$, where $\mathbf{e}_k \in \mathcal{R}^{d_e}$. GZSL aims at classifying instances from both source or target classes, i.e., $f_{gzsl} : \mathcal{X} \rightarrow \mathcal{Y}_\mathcal{S} \cup \mathcal{Y}_\mathcal{T}$. In this paper, we apply GZSL in a transductive way, assuming that unlabeled target data is provided in training [12].

3.2. Overview

As shown in Figure 2, our Self-supervised Domain-aware Generative Network (SDGN) is composed of two modules. (1) The cross-domain feature generating module is responsible for generating features for source and target domains. (2) The self-supervised learning module consists of reference anchor learning, domain feature reconstruction, and cross-domain triplet mining. The details are as follows.

3.3. Cross-Domain Feature Generating Module

In this paper, we learn a generative model that can synthesize fake visual features for any class of interest, purely based on the attribute vector of the class, which can be leveraged in training a standard supervised classifier for GZSL.

Given the training images of source domain and unlabeled images of target domain, our goal is to learn a generator $G : \mathcal{E} \times \mathcal{Z} \rightarrow \mathcal{X}$, which takes a class embedding $\mathbf{e}^y \in \mathcal{E}$ and a Gaussian noise $z \in \mathcal{Z}$ as inputs, and generates a visual feature $\tilde{x} \in \mathcal{X}$. The discriminator $D_s : \mathcal{X} \times \mathcal{E} \rightarrow [0, 1]$ is designed for source domain, which takes a real feature x^s or a synthetic feature $\tilde{x}^s$ with the associated class embedding $\mathbf{e}^s$ as input. The aim of D_s is to discern whether the source feature and $\mathbf{e}^s$ are matched. The generator G intends to confuse D_s by producing features highly correlated with $\mathbf{e}^s$ by a Wasserstein adversarial loss [3]:

$$\begin{aligned}\mathcal{L}^s_{WGAN} = \min_{G} \max_{D_s} \mathbb{E}[D_s(x^s, \mathbf{e}^s)] - \mathbb{E}[D_s(\tilde{x}^s, \mathbf{e}^s)] \\ -\lambda \mathbb{E}[(\|\nabla_{\hat{x}^s} D_s(\hat{x}^s, \mathbf{e}^s)\|_2 - 1)^2],\end{aligned} \tag{1}$$

where the third term is gradient penalty, $\hat{x}^s = \alpha x^s + (1 - \alpha)\tilde{x}^s, \alpha \sim U(0, 1)$.

The discriminator D_s can only model the distribution of source data. However, the appearance variance between different domains can bring in domain bias that causes G to produce features that are similar to source data even in target scenario. Therefore, we propose a target domain discriminator D_t to capture characteristics specifically for unseen classes. D_t takes image feature pairs as input, which are composed of a visual feature and its corresponding multi-label $\mathcal{M}^t$. Here, the multi-label $\mathcal{M}^t$ will be discussed in detail in Section 3.4. It represents the reconstructed feature in a shared space. The purpose of D_t is to judge whether the input pair belongs to target domain. By playing a min-max game between D_t and G, $\tilde{x}^t$ and its reconstructed feature will be forced to be remained in target domain, which ensures the domain label consistency.

$$\begin{aligned}\mathcal{L}^t_{WGAN} = \min_{G} \max_{D_t} \mathbb{E}[D_t(x^t_i, \mathcal{M}^{t'}_i)] - \mathbb{E}[D_t(\tilde{x}^t_j, \mathcal{M}^t_j)] \\ -\lambda \mathbb{E}[(\|\nabla_{\hat{x}^t} D_t(\hat{x}^t, \hat{\mathcal{M}}^t)\|_2 - 1)^2],\end{aligned} \tag{2}$$

where $\mathcal{M}^{t'}_i$ and $\mathcal{M}^t_j$ are the multi-label of real feature x^t_i and fake feature $\tilde{x}^t_j$. The third term is the gradient penalty, $\hat{x}^t = \alpha x^t + (1 - \alpha)\tilde{x}^t$ and $\hat{\mathcal{M}}^t = \alpha \mathcal{M}^{t'}_i + (1 - \alpha)\mathcal{M}^t_j$.

To make sure that $\tilde{x}^s$ can well suit the final classification task, we expect them to be predicted correctly by the a pre-trained classifier C with a loss defined as in Eq. (3).

$$\mathcal{L}_C = -\mathbb{E}_{(\tilde{x}^s, y^s) \sim P_{\tilde{x}^s}} [logP(y^s | \tilde{x}^s, \theta_C)], \tag{3}$$

where $P(y^s | \tilde{x}^s, \theta_C)$ is the classification probability and θ_C denotes fixed parameters of the pre-trained classifier.

3.4. Self-supervised Learning Module

Despite that the cross-domain feature generating module is powerful in synthesizing high-quality features, it still fails to consider the relations between source and target domains, and does not differentiate seen classes from unseen classes. Therefore, we propose reference anchors for building a shared space for seen and unseen classes, where we exploit the domain relationships by a novel **Self-supervised Learning Module (SLM)**. The details are as follows.

Reference Anchor Learning. We introduce reference anchors to bridge seen and unseen classes and use them to reconstruct features in a joint embedding space. How to choose anchors is of vital importance. We notice that each class can be represented as a attribute vector, where each dimension encodes a high-level visual property. This gives rise to the idea of training an attribute classifier and extracting its parameters as anchors. Since the attribute classification is achieved by the dot production between linear weights and high-level features, these parameters encode visual attributes that are universal for the source and target domains, thus are promising in bridging two domains. The similar idea of constructing anchors can be found in [47]. For a series of attribute classifiers $\{g_1, g_2, \cdots, g_{d_e}\}$ w.r.t. d_e attributes, we extract their weight parameters as anchors: $\{A_1, A_2, \cdots, A_{d_e}\}$, $A_i \in \mathcal{R}^{d_v}$. In the end-to-end training process, these anchors will be dynamically updated to better relate source classes with target classes.

Domain Feature Reconstruction. Based on the anchors, we could reconstruct synthetic features for each domain. By comparing synthesized features with anchors, we obtain visual similarities as the soft multi-label. The multi-label function is defined in softmax form: $\mathcal{M}^{(k)}_i = \frac{exp(<A_k, \tilde{x}_i>)}{\sum_j exp(<A_j, \tilde{x}_i>)}$. $\mathcal{M}^{(k)}_i$ denotes the similarity between i-th image and k-th anchor, and $< \cdot >$ is the cosine similarity. Then, each reconstructed feature can be interpreted as a convex combination of anchors, with $\mathcal{M}_i$ as coefficients:

$$x^r_i = \sum_{k=1}^{d_e} \mathcal{M}^{(k)}_i A_k, \tag{4}$$

where x^r_i denotes the reconstructed feature of the i-th image. These features are lied in the common space spanned by anchors, and can be utilized to investigate the relationship of different domains.

Cross-Domain Triplet Mining. To explore the relations between distributions of source and target domains, our idea is intuitive as follows. The instances of source classes should be pushed away from those of target classes. This idea presents a natural self-supervised signal that the similarity of reconstructed features from different domains should be much lower than the similarity between reconstructed features and their synthesized features. Thus the reconstructed features from different domains form a negative pair $[x^{rs}, x^{rt}]_{neg}$. During training, each iteration samples a batch source and a batch target attributes to produce features that can be combined as negative pairs. Naturally, the synthesized feature $\tilde{x}$ and its reconstructed feature x^r form a positive pair $[\tilde{x}, x^r]_{pos}$. Totally, we obtain two kinds of cross-domain triplets: $[\tilde{x}^t, x^{rt}, x^{rs}]$ and $[\tilde{x}^s, x^{rs}, x^{rt}]$. Guided by the self-supervised signal, we design a cross-domain triplet loss to mine the information of triplets for target and source domain respectively:

$$\mathcal{L}_{cross}^{T} = \max(\|x^{rt} - \tilde{x}^t\|^2 - \|x^{rt} - x^{rs}\|^2 + \mu, 0), \quad (5)$$

$$\mathcal{L}_{cross}^{S} = \max(\|x^{rs} - \tilde{x}^s\|^2 - \|x^{rs} - x^{rt}\|^2 + \mu, 0), \quad (6)$$

where the threshold μ is a margin. Consequently, the final cross-domain triplet loss is:

$$\mathcal{L}_{cross} = \mathcal{L}_{cross}^{T} + \mathcal{L}_{cross}^{S}. \quad (7)$$

Noticeably, the cross-domain triplet mining is completely free of target ground truth since we utilize the self-supervisory information inferred from relations between domains. Constrained by the cross-domain triplet loss, we achieve the domain-level discrimination power that benefits the classification for both source and target classes.

3.5. Model Training and Prediction

The final loss objective $\mathcal{L}$ aggregates the WGAN loss, classification loss and cross-domain triplet loss:

$$\mathcal{L} = \mathcal{L}_{WGAN}^{s} + \lambda_t \mathcal{L}_{WGAN}^{t} + \lambda_a \mathcal{L}_{cross} + \lambda_c \mathcal{L}_{C}. \quad (8)$$

where $\lambda_t, \lambda_a, \lambda_c$ are loss coefficients. Once the generator G is well-trained, we leverage it to produce samples for any classes of interest based on class embedding $\mathbf{e}^y$.

With sufficient samples in hand, we transform GZSL into a standard classification model. Specifically, we combine the synthesized target features $\tilde{x}^t$ and real source features x^s to construct the training set. Then we train an ultimate softmax classifier by minimizing the negative log likelihood loss: $\min_\theta -\frac{1}{|\mathcal{X}|}\sum_{(x,y)\in(\mathcal{X},\mathcal{Y})} \log P(y|x,\theta)$, where $P(y|x,\theta) = \frac{\exp(\theta_y^T x)}{\sum_{j=1}^{|\mathcal{Y}|}\exp(\theta_j^T x)}$ is the classification probability and θ denotes classifier parameters. The final prediction result is derived by $f(x) = \arg\max_y P(y|x,\theta)$.

3.6. Discussions

In this section, we discuss the differences between SDGN and three relevant methods including CEWGAN-OD [22], f-VAEGAN[45] and SABR-T [27]. (1) CEWGAN-OD uses a domain detector to determine the domain type before the final classification. Our method shares the similar idea in separating the seen and unseen classes to reduce domain bias. However, CEWGAN-OD needs additional classifiers, while our method directly constructs a joint discriminative embedding that is aware of domain boundary. (2) F-VAEGAN and SABR-T utilize two discriminators for source and target domains respectively, which are similar to SGDN. Both of them attempt to learn the target data distribution. However, they cannot train a target discriminator with supervised information like source domain, while our method uses multi-labels obtained from anchors acted as conditions for the target domain discriminator. (3) Compared to the above methods, our SDGN performs the self-supervised learning on synthesized data to explore the distribution of two domains and their relations.

4. Experiments

In this section, we conduct extensive experiments to evaluate our SDGN. Please refer to the **Supplementary Material** for training details and more experimental results.

4.1. Datasets and Evaluation Metrics

Datasets. We evaluate our model on five challenging datasets including CUB [39], AwA1 [15], AwA2 [44], SUN [26], and FLO [24]. CUB includes 11K images and 200 species of birds labeled with 312-D attributes. AwA1 and AwA2 consist of 50 kinds of animals described by 85-D attributes, containing 30K and 37K images respectively. SUN is a large-scale scene attribute dataset, including 717 classes and 14K images with 102-D attributes. FLO consists of 8K images from 102 flower classes. Among these datasets, AwA1 and AwA2 are corase-grained while others are fine-grained.

Evaluation Metrics. In GZSL, since testing instances come from either seen or unseen classes, the search space takes all classes into account. We adopt the average per-class top-1 accuracy $ACA_{\mathcal{S}}$ and $ACA_{\mathcal{U}}$ to evaluate seen and unseen classes respectively. To measure the comprehensive performance, we use the harmonic mean as final metrics: $H = \frac{2\times ACA_{\mathcal{S}}\times ACA_{\mathcal{U}}}{ACA_{\mathcal{S}}+ACA_{\mathcal{U}}}$.

4.2. Implementation Details

Similar to Xian *et al.* [42], we use the pretrained ResNet-101 to extract CNN features for real images. The generator (G) and discriminators (D_s and D_t) in SDGN are all implemented via the multilayer perceptron. The source features concatenated with $\mathbf{e}^y$ are fed into D_s while the input of D_t is the feature pairs composed of target features and their multi-labels. Here, we do not directly use reconstructed feature x^{rt} as input. Since $\mathcal{M}^t$ can be considered as coefficients in the shared space spanned by anchors, and they can represent the same information as x^{rt}. Furthermore, the multi-label is low-dimensional and can reduce the model size and computational cost. The coefficients λ_t, λ_a and λ_c in Eq. (8) are set as $1, 0.1$ and 0.01. As for obtaining anchors, since all the GZSL datasets provide attributes for each class, we assign each image with its attribute label of the corresponding class. We train a simple multi-label learning model for attribute classification and extract the weights of the final fully-connected layer as anchors.

Table 1. Comparison of our method with the state-of-the-art inductive (**I**) and transductive (**T**) methods. We measure top-1 accuracy (**T1**) in ZSL setting. In GZSL, we report top-1 accuracy for seen classes (**S**) and unseen classes (**U**). **H** denotes the harmonic mean. Red font and blue font denote the highest and the second highest results.

	Method	**Zero-shot Learning**					**Generalized Zero-shot Learning**														
		CUB	**AwA1**	**AwA2**	**SUN**	**FLO**	**CUB**			**AwA1**			**AwA2**			**SUN**			**FLO**		
		T1	**T1**	**T1**	**T1**	**T1**	**S**	**U**	**H**	**S**	**U**	**H**	**S**	**U**	**H**	**S**	**U**	**H**	**S**	**U**	**H**
I	**ALE** [1]	54.9	59.9	62.5	58.1	48.5	63.8	23.7	34.4	76.1	16.8	27.5	81.8	14.0	23.9	33.1	21.8	26.3	61.6	13.3	21.9
	SYNC [4]	55.6	54.0	46.6	56.3	-	70.9	11.5	19.8	87.3	8.9	16.2	90.5	10.0	18.0	43.3	7.9	13.4	66.3	7.4	13.3
	SJE [2]	53.9	65.6	61.9	53.7	53.4	59.2	23.5	33.6	74.6	11.3	19.6	73.9	8.0	14.4	30.5	14.7	19.8	47.6	13.9	21.5
	ESZSL [29]	53.9	58.2	58.6	54.5	51.0	63.8	12.6	21.0	75.6	6.6	12.1	77.8	5.9	11.9	27.9	11.0	15.8	56.8	11.4	19.0
	LATEM [41]	49.3	55.1	55.8	55.3	40.4	57.3	15.2	24.0	71.7	7.3	13.3	77.3	11.5	20.0	28.8	14.7	19.5	47.6	6.6	11.5
	DeViSE [11]	52.0	54.2	59.7	56.5	45.9	53.0	23.8	32.8	68.7	13.4	22.4	74.7	17.1	27.8	27.4	16.9	20.9	44.2	9.9	16.2
	DEM [48]	51.7	68.4	67.2	61.9	-	54.0	19.6	13.6	32.8	84.7	47.3	86.4	30.5	45.1	25.6	34.3	20.5	-	-	-
	SP-AEN [6]	55.4	-	58.5	59.2	-	70.6	34.7	46.6	-	-	-	90.9	23.3	37.1	38.6	24.9	30.3	-	-	-
	f-CLSWGAN [43]	57.3	68.2	-	60.8	67.2	57.7	43.7	49.7	61.4	57.9	59.6	68.9	52.1	59.4	36.6	42.6	39.4	73.8	59.0	65.6
	CADA-VAE [31]	-	-	-	-	-	53.5	51.6	52.4	72.8	57.3	64.1	75.0	55.8	63.9	35.7	47.2	40.6	-	-	-
	LisGAN [17]	58.8	70.6		61.7	69.6	57.9	46.5	51.6	76.3	52.6	62.3	-	-	-	37.8	42.9	40.2	83.8	57.7	68.3
T	**GFZSL** [38]	50.0	48.1	78.6	64.0	85.4	45.8	24.9	32.2	67.2	31.7	43.1	-	-	-	-	-	-	75.0	21.8	33.8
	DSRL [46]	48.7	74.7	72.8	56.8	57.7	39.0	17.3	24.0	74.7	20.8	32.6	-	-	-	25.0	17.7	20.7	64.3	26.9	37.9
	GMN [30]	64.6	82.5	-	64.3	-	70.6	60.2	65.0	79.2	70.8	74.8	-	-	-	40.7	57.1	47.5	-	-	-
	f-VAEGAN [45]	71.1	-	89.8	70.1	89.1	65.1	61.4	63.2	-	-	-	88.6	84.8	86.7	41.9	60.6	49.6	87.2	78.7	82.7
	GXE [18]	61.3	89.8	83.2	63.5	-	68.7	57.0	62.3	89.0	87.7	88.4	90.0	80.2	84.8	58.1	45.4	51.0	-	-	-
	SDGN	74.9	92.3	93.4	68.4	81.8	70.2	69.9	70.1	88.1	87.3	87.7	89.3	88.8	89.1	46.0	62.0	52.8	91.4	78.3	84.4

4.3. Comparison with the state-of-the-art.

In this experiment, we compare our proposed SDGN with several state-of-the-art methods.

Compared Methods. We divide GZSL methods into two types including inductive methods that only utilize the labeled source data, and transductive methods that assume the access of unlabeled target data. (1) In the inductive setting, we select several competitive methods including ALE [1], SYNC [4], SJE [2], ESZSL [29], LATEM [41], DeViSE [11], DEM [48], SP-AEN [6], f-CLSWGAN [43], CADA-VAE [31], and LisGAN [17] (2) In the transductive setting, we choose five state-of-the-art methods including GFZSL [38], DSRL [46], GMN [30], f-VAEGAN [45], and GXE [18]. Our model falls into the transductive type. Here we do not include QFSL [35] and SABR-T [27], since they refine the visual features, while most methods adopt off-the-shelf features extracted by the pre-trained ResNet101 following Xian *et al.* [44]. Direct comparison of their performance with other methods and ours is not fair.

Generalized Zero-shot Learning Results. As shown in Table 1, SDGN achieves the state-of-the-art results on five datasets in competition with both inductive methods and transductive methods. Our method either ranks the first or the second among most of results of seen accuracy, unseen accuracy, and harmonic mean. Based on the results, we have the following observations. (1) In the inductive setting, our method achieves significantly higher results than all the previous works. Even compared with the best inductive results, SDGN attains a huge accuracy boost as high as 17.7% on CUB, 23.6% an AwA1, 25.2% on AwA2, 12.2% on SUN, and 16.1% on FLO. This can be ascribed to the effective utilization of unlabeled data in SDGN. D_t can well align the synthesized data with the authentic distribution of target domain, while SLM can exploit the information lurking in the relation between source and target domains. Notably, generative methods (f-CLSWGAN, CADA-VAE, and LisGAN) perform better than cross-modal mapping methods (ALE, SJE, ESZSL, etc.). The cross-modal mapping methods get high scores on seen accuracy but perform poorly on unseen accuracy and harmonic mean. This indicates that they are overfitted on the seen classes, while generative models synthesize features which straightly contribute to a classifier designed for all classes. (2) In the transductive setting, our method acquires the excellent performance and establishes the new state-of-art results on four datasets, i.e. 70.1% on CUB, 89.1% on AwA2, 52.8% on SUN and 84.4% on FLO. Surprisingly, on CUB dataset, we surpass the second highest result by a significant margin of 5.1%. The reason is that CUB is a very fine-grained dataset, consisting of different species of birds with small variances, and the features of different domains are close to each other. SDGN can effectively address this problem by using the SLM to separate source domain from target domain. On AwA1 dataset, our method performs favorably against GXE, i.e. 87.7% vs 88.4%. However, our method consistently outperforms GXE on other datasets. Especially on fine-grained datasets CUB and SUN, we surpass GXE by 7.8% and 1.8%. This is because that GXE merely uses unlabeled data to slightly calibrate the model that is already trained on source data, while SDGN essentially discerns the variances between classes from different domains.

Conventional Zero-shot Learning Results. We also conduct experiments on conventional ZSL. As shown in Table 1, our method presents a significant boost in accuracy compared with the state-of-the-art, i.e., 3.8% on CUB, 2.5% on AwA1 and 3.6% on AwA2. On the large-scale dataset SUN, our model also achieves the second highest result. We attribute this high performance gain to the reduced bias in the model by the self-supervised learning to make a clear decision boundary between seen and unseen classes.

4.4. Ablation Study

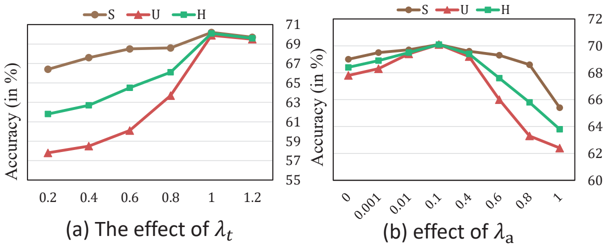
To show further insights about SDGN, we perform ablation study to evaluate the effect of different model components and different values of loss coefficients.

Analysis of Model Components. We perform detailed analysis on the benefits of different modules of SDGN. We take three datasets CUB, AwA2 and SUN as examples. The baseline model consists of a generator G and a single discriminator D_s. Based on the baseline, we test the model performance by adding D_t and SLM. As shown in Table 2, we show significant improvements over the baseline. The complete version of SDGN gives the highest results on all datasets, achieving a whopping accuracy gain of 24.6%, 36.8% and 19.4% on unseen accuracy, and 19.1%, 29.1% and 13.4% on harmonic mean, which proves the effectiveness of SDGN. The results are analyzed as follows: (1) The introduction of D_t remarkably enhances the performance on harmonic mean by a large margin, i.e., 17.5% on CUB, 26.9% on AwA2, and 10.8% on SUN. This improvement can be mainly ascribed to the strong ability of D_t in building reliable data manifold for target domain, since it restricts the synthesized and reconstructed feature in the correct domain. Meanwhile, we also observe a considerable improvement on the seen accuracy. This is because that reliable target samples reduce the introduced noises imposed on the ultimate softmax classifier, which is also beneficial for seen class recognition. (2) After utilizing SLM, the model performance takes further improvement. SLM contributes to the essential accuracy enhancement. The reason is that the cross-domain triplet mining disentangles the target domain distribution from source domain. This leads to a more accurate decision boundary between source and target classes, which enhances the comprehensive recognition ability for both seen classes and unseen classes. However, on AwA2 dataset, the seen accuracy has a slight drop of 1.9%, and the results of $(G + D_s + D_t)$ exhibit the strong bias towards seen classes, i.e., 91.2% (S) VS 83.0% (U). This is because AwA2 is a coarse-grained and small-scale dataset with only 50 animal classes. It is less-challenging and the seen accuracy can easily reach the peak. Therefore, SLM can hardly influence the seen accuracy. Here, we do not test the performance of ($G + D_s +$ SLM), since without D_t, the synthesized target features contain too much noises, and it is difficult to separate target domain from source domain. Therefore, it does not make sense to adopt SLM alone to separate the synthesized features without D_t.

Analysis of Loss Coefficients. We study the effect of loss coefficients λ_t and λ_a in Eq. (8) to obtain a more intuitive observation on the module influence. We take CUB dataset as an example. λ_t and λ_a controls the importance of D_t and SLM. The results on CUB dataset are shown in Figure 3. At first, as λ_t grows, unseen accuracy and harmonic mean gain consistent improvements, which indicates that keeping stressing on D_t will generate more and more reliable synthetic features. The accuracy reaches the peak at $\lambda_t = 1.0$, and then decreases as λ_t grows. This means the best configuration of the cross-domain feature generating module is to give the same attention on D_s and D_t. Putting imbalanced weights causes training instability, thus impairs the model performance. As for λ_a, in the early stage, the increasing of λ_a leads to the accuracy improvements on all of the three indicators (S, U, H), which indicates that putting more weight on SLM can acquire more discriminative power between source and target domains, which maximizes the comprehensive classification competence for all classes. The accuracy reaches the peak at $\lambda_a = 0.1$, and then decreases as λ_a grows. That is because weighing too much on SLM might hurt the training stability of feature generation module due to the introduced noises.

Table 2. Ablation results on CUB, AwA2 and SUN datasets.

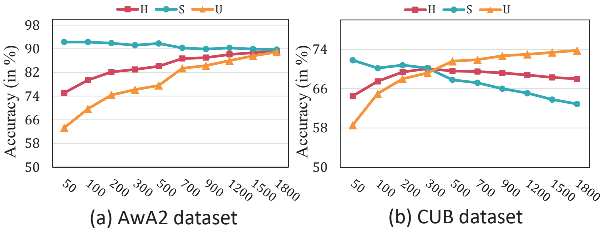
G	D_s	D_t	SLM	CUB			AwA2			SUN		
				S	U	H	S	U	H	S	U	H
✓	✓	✗	✗	58.2	45.3	51.0	67.5	54.0	60.0	36.6	42.6	39.4
✓	✓	✓	✗	69.0	68.0	68.5	91.2	83.0	86.9	44.0	58.5	50.2
✓	✓	✓	✓	**70.2**	**69.9**	**70.1**	89.3	**88.8**	**89.1**	**46.0**	**62.0**	**52.8**



(a) The effect of λ_t

(b) effect of λ_a

Figure 3. Comparison results of coefficients λ_t and λ_a on CUB.



(a) AwA2 dataset

(b) CUB dataset

Figure 4. Results of synthesized feature number per class.

4.5. Analysis of Number of Synthesized Samples

We evaluate the impact of the number of synthetic samples, denoted as N_{syn}. As shown in Figure 4, we observe that the best N_{syn} is 1800 for AwA2 and 300 for CUB. On AwA2, as N_{syn} increases, the unseen accuracy and har-

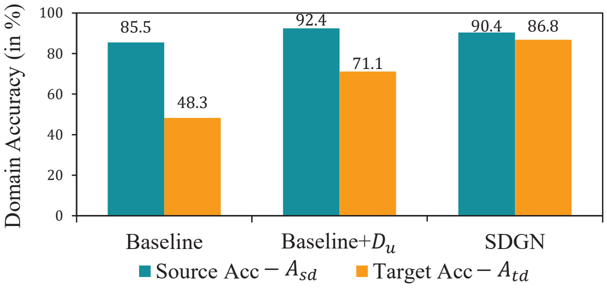
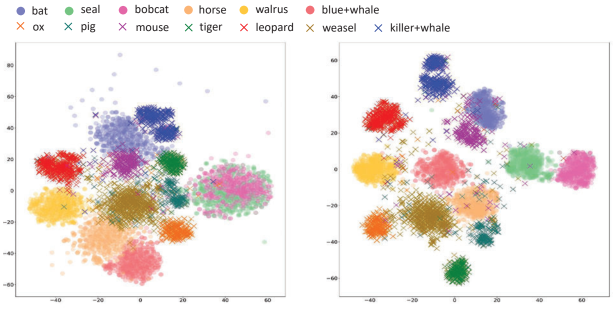


Figure 5. Comparison of the source domain accuracy A_{sd} and the target domain accuracy A_{td}.



(a) Baseline synthesized features

(b) SDGN synthesized features

Figure 6. The t-SNE visualization of (a) Baseline synthesized features and (b) SDGN synthesized features. ● denotes features of target domain and × denotes features of source domain.

monic mean exhibit a rapid growing trend. At first, the seen accuracy is much higher than unseen accuracy, implying that synthetic target samples are not sufficient. Synthesizing more samples is effective for constructing the discrimination power for unseen classes. Also, we can see that the highest harmonic mean is achieved when the seen and unseen accuracy attain a similar level, i.e., reach the best balance. In addition, we observe seen accuracy drops a little when increasing the synthetic features. This is reasonable since the bias towards source domain is alleviated. Thus a raise in the unseen accuracy is likely to cause a drop in the seen accuracy. The key is how to trade off between seen and unseen accuracies. On CUB, the harmonic mean first increases as N_{syn} grows larger, but decreases after reaching the peak value at the point of 300. The reason is that there exists an upper bound of the synthetic diversity. Thus when N_{syn} reaches a certain number, more features make no contribution for improving classification.

4.6. Quantitative Examination of Domain Bias

In this experiment, we examine the effectiveness of SDGN in reducing the bias towards seen classes. To quantify the domain bias, we follow the experimental setting proposed in [22], where all of the unseen classes are regarded as a single target class, and all of the seen classes are considered as a single source class. The images are gone through a binary classifier to determine whether they belong to target or source domains. We use A_{td} and A_{sd} to denote the target domain accuracy and the source domain accuracy, respectively. As visualized in Figure 5, the baseline presents an extremely unbalanced domain accuracy on CUB, with A_{sd} exceeding 85% while A_{td} is only 48.3%. The huge domain accuracy gap implies that the bias towards seen classes is very severe. Noticeably, the utilization of D_t strongly improves the A_{td} by 22.8%, and A_{sd} by 6.9%. Surprisingly, the domain accuracy gap decreases from 37.2% to 21.3%. Adding SLM further eliminates the domain bias to the maximum degree. A_{td} is raised by 15.7% and the domain gap is reduced to 3.6%. These results manifest the excellent efficacy of our SDGN in mitigating the strong bias by making a balance between seen and unseen accuracy. At a modest expense of A_{sd}, SDGN significantly advances the generalization ability for unseen classes.

4.7. Visualization

To further show the effectiveness of SDGN, we conduct the t-SNE [21] visualization for the synthesized features of the baseline and SDGN on AwA1. As shown in Figure 6, the features produced by baseline are tightly crowding together, and a bunch of target features are close to or even overlap source features, e.g., “mouse” features (purple crosses) and “bat” features (purple dots). While the synthesized target features of SDGN are evidently pushed away from source features, e.g., “bat” features (purple dots) and “killer+whale” features (blue crosses). Moreover, features have more segregated clusters for both source and target domains. Synthesized samples in the same category become more compact. This verifies that SDGN is powerful in disentangling feature distributions of different domains, and constructing more discriminative representations.

5. Conclusions

In this paper, we propose a Self-supervised Domain-aware Generative Network for GZSL. We explore self-supervised learning in the feature generating model, with anchors acting as a bridge between seen and unseen classes. Based on the multi-label derived from anchors, we conduct a cross-domain triplet mining to exploit the cross-domain relations. Experiments show the effectiveness. In the future, we will apply our model for visual tracking [50, 51].

6. Acknowledgment

This work was partially supported by the National Key Research and Development Program under Grant No. 2018YFB0804204, Strategic Priority Research Program of Chinese Academy of Sciences (No. XDC02050500), National Nature Science Foundation of China (NSFC) under Grants 61751211, 61972020, 61532009, 61622211, U19B2038 and 61620106009, Open Project Program of the National Laboratory of Pattern Recognition (NLPR) under Grant 201700022, and Youth Innovation Promotion Association CAS 2018166.

References

[1] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for attribute-based classification. In *CVPR*, 2013.

[2] Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. Evaluation of output embeddings for fine-grained image classification. In *CVPR*, 2015.

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[4] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Synthesized classifiers for zero-shot learning. In *CVPR*, 2016.

[5] Weilun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *ECCV*, 2016.

[6] Long Chen, Hanwang Zhang, Jun Xiao, Wei Liu, and Shih-Fu Chang. Zero-shot visual recognition using semantics-preserving adversarial embedding networks. In *CVPR*, 2018.

[7] Mohamed Elhoseiny and Mohamed Elfeki. Creativity inspired zero-shot learning. In *ICCV*, 2019.

[8] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[9] Rafael Felix, Vijay BG Kumar, Ian Reid, and Gustavo Carneiro. Multi-modal cycle-consistent generalized zero-shot learning. In *ECCV*, 2018.

[10] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *CVPR*, 2017.

[11] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *NeurlPS*, 2013.

[12] Yanwei Fu, Timothy M Hospedales, Tao Xiang, and Shaogang Gong. Transductive multi-view zero-shot learning. *IEEE transactions on pattern analysis and machine intelligence*, 37(11):2332–2345, 2015.

[13] He Huang, Changhu Wang, Philip S Yu, and Chang-Dong Wang. Generative dual adversarial network for generalized zero-shot learning. In *CVPR*, 2019.

[14] Vinay Kumar Verma, Gundeep Arora, Ashish Mishra, and Piyush Rai. Generalized zero-shot learning via synthesized examples. In *CVPR*, 2018.

[15] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2013.

[16] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *ECCV*, 2016.

[17] Jingjing Li, Mengmeng Jin, Ke Lu, Zhengming Ding, Lei Zhu, and Zi Huang. Leveraging the invariant side of generative zero-shot learning. *arXiv preprint arXiv:1904.04092*, 2019.

[18] Kai Li, Martin Renqiang Min, and Yun Fu. Rethinking zero-shot learning: A conditional visual classification perspective. In *CVPR*, 2019.

[19] Jiawei Liu, Zheng-Jun Zha, Di Chen, Richang Hong, and Meng Wang. Adaptive transfer network for cross-domain person re-identification. In *CVPR*, 2019.

[20] Shichen Liu, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Generalized zero-shot learning with deep calibration network. In *Advances in Neural Information Processing Systems*, 2018.

[21] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[22] Devraj Mandal, Sanath Narayan, Sai Kumar Dwivedi, Vikram Gupta, Shuaib Ahmed, Fahad Shahbaz Khan, and Ling Shao. Out-of-distribution detection for generalized zero-shot action recognition. In *CVPR*, 2019.

[23] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[24] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

[25] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

[26] Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, 2012.

[27] Akanksha Paul, Narayanan C Krishnan, and Prateek Munjal. Semantically aligned bias reducing zero shot learning. In *CVPR*, 2019.

[28] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *CVPR*, 2016.

[29] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, 2015.

[30] Mert Bulent Sariyildiz and Ramazan Gokberk Cinbis. Gradient matching generative networks for zero-shot learning. In *CVPR*, 2019.

[31] Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero-and few-shot learning via aligned variational autoencoders. In *CVPR*, 2019.

[32] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *IEEE International Conference on Robotics and Automation*, pages 1134–1141, 2018.

[33] Vivek Sharma, Makarand Tapaswi, M Saquib Sarfraz, and Rainer Stiefelhagen. Self-supervised learning of face representations for video face clustering. *arXiv preprint arXiv:1903.01000*, 2019.

[34] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *NeurlPS*, 2013.

[35] Jie Song, Chengchao Shen, Yezhou Yang, Yang Liu, and Mingli Song. Transductive unbiased embedding for zero-shot learning. In *CVPR*, 2018.

[36] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018.

[37] Yao-Hung Hubert Tsai, Liang-Kang Huang, and Ruslan Salakhutdinov. Learning robust visual-semantic embeddings. In *ICCV*, 2017.

[38] Vinay Kumar Verma and Piyush Rai. A simple exponential family framework for zero-shot learning. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 792–808. Springer, 2017.

[39] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[40] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019.

[41] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *CVPR*, 2016.

[42] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[43] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *CVPR*, 2018.

[44] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning-the good, the bad and the ugly. In *CVPR*, 2017.

[45] Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. f-vaegan-d2: A feature generating framework for any-shot learning. In *CVPR*, 2019.

[46] Meng Ye and Yuhong Guo. Zero-shot classification with discriminative semantic representation learning. In *CVPR*, 2017.

[47] Hong-Xing Yu, Wei-Shi Zheng, Ancong Wu, Xiaowei Guo, Shaogang Gong, and Jian-Huang Lai. Unsupervised person re-identification by soft multilabel learning. In *CVPR*, 2019.

[48] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. In *CVPR*, 2017.

[49] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, 2017.

[50] Tianzhu Zhang, Si Liu, Changsheng Xu, Bin Liu, and Ming-Hsuan Yang. Correlation particle filter for visual tracking. *IEEE Transactions on Image Processing*, 27(6):2676–2687, 2018.

[51] Tianzhu Zhang, Changsheng Xu, and Ming-Hsuan Yang. Learning multi-task correlation particle filters for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):365–378, 2019.