Generative Model with Semantic Embedding and Integrated Classifier for Generalized Zero-Shot Learning

Ayyappa Kumar Pambala
Indian Institute of Science
ayyappap@iisc.ac.in

Titir Dutta
Indian Institute of Science
titird@iisc.ac.in

Soma Biswas
Indian Institute of Science
somabiswas@iisc.ac.in

Abstract

Generative models have achieved impressive performance for the generalized zero-shot learning task by learning the mapping from attributes to feature space. In this work, we propose to derive semantic inferences from images and use them for the generation, which enables us to capture the bidirectional information i.e., visual to semantic and semantic to visual spaces. Specifically, we propose a Semantic Embedding module which not only gives image specific semantic information to the generative model for generation of better features, but also makes sure that the generated features can be mapped to the correct semantic space. We also propose an Integrated Classifier, which is trained along with the generator. This module not only eliminates the requirement of additional classifier for new object categories which is required by the existing generative approaches, but also facilitates the generation of more discriminative and useful features. This approach can be used seamlessly for the task of few-shot learning. Extensive experiments on four benchmark datasets, namely, CUB, SUN, AWA1, AWA2 for both zero-shot learning and few-shot setting show the effectiveness of the proposed approach.

1. Introduction

Zero-shot learning (ZSL) [1][2][4][16] addresses the problem of image classification of previously unseen classes, which is a very relevant real-world problem, since the number of object classes is dynamically increasing. A more generalized scenario, where the a-priori knowledge of whether the object is seen or unseen is not available, is addressed as generalized ZSL (GZSL) [34][35][19][24]. These approaches generally utilize the semantic descriptions (manual attributes, word2vec, etc.) of both the seen and unseen classes to bridge the gap between them. Traditional approaches [1][23][16][2][15][4] learn the transformations from the image-space to semantic space or vice-versa in order to classify objects from unseen classes. Re-

cent advances in deep learning [19][35][29][36][24] address this problem by generating synthetic image features for the unseen classes for which no training data is available. A variant of ZSL and GZSL is Few-Shot Learning (FSL) [11][25][31] and FSL without forgetting [8], where the task is to classify image samples from classes, for which very few samples are available for training.

In this work, we propose a general framework to address these problems by augmenting a generative model with two novel modules, namely *Semantic Embedding* (SE) and *Integrated Classifier* (IC) to further enhance its performance. In general, a generative model learns the mapping from semantic to the visual space [35][36][29][6]. In contrast, in this work, we propose to incorporate bidirectional embeddings (i.e. visual to semantic and semantic to visual) to improve the quality of generated image features. The SE module, in addition to giving image specific semantic information to the generative model for better feature generation, also ensures that the generated features can be mapped to the correct semantic space.

For majority of generative approaches, the generated features are used to train a classifier, thus transforming the ZSL task to a classical supervised classification problem. This requires the classifier to be trained each time a new class is encountered. Our second contribution is the IC module, which is trained simultaneously with the generator. It is designed to take a pair of image features as input, and identify whether they belong to the same or different classes. Since the real images are not available for unseen classes, we propose to utilize the generated image samples from unseen classes to incorporate the unseen class-discrimination ability into the classifier. This helps in reducing the bias towards the seen classes. The main contributions in this work are summarized below.

1. We propose a bidirectional mapping using generative adversarial networks between the visual space and semantic space, to improve the quality of synthetic image features.

2. We propose a classifier which is trained along with

the generator and can be used directly during testing, thereby eliminating the need of retraining a separate classifier.

3. Our proposed approach can be extended seamlessly to few-shot learning setting as well.

4. Extensive experiments on CUB, SUN, AWA1 and AWA2 show the effectiveness of the framework.

The rest of the paper is organized as follows. The related works and the proposed framework are described in Section 2 and Section 3 respectively. Results of extensive evaluation are reported and analyzed in Section 4 and the paper ends with a summary in Section 5.

2. Related work

There has been significant amount of work in the area of ZSL and FSL reported in the recent literature. In this section, we give pointers to the work closely related to the proposed work.

Zero-shot Learning In this work, we address both ZSL and GZSL problems. Early works [7][1][2] learn a linear compatibility function between the visual space and semantic embedding space. [15] finds a linear mapping from visual to semantic space with a reconstruction objective which maps back to visual space; whereas [33][26] learn non-linear compatibility functions from visual to semantic space. [30] is a generative model which formulates the class conditional distributions as Gaussians.

Advances in generative models [9][10][6][17] have enabled the generation of new synthetic images to mimic a particular input data distribution. Inspired by the these approaches, recent algorithms address the problem of lack of training data for the unseen classes by generating synthetic images for the same. CVAE-ZSL [19] uses conditional VAE to generate such synthetic features. SE-GZSL [29] and [6] use VAE and GAN respectively with regressor as a feedback on the generated features to improve the feature generation. f-CLSWGAN [35] uses GAN to generate features for unseen classes. f-VAEGAN [36] utilizes both VAE and GAN strengths to improve the generation quality.[13] proposes to learn class prototypes by aligning visual and semantic space simultaneously using coupled dictionary learning approach. Different from these approaches, ReViSE [28] and CADA-VAE [24] propose to learn a common intermediate space from both visual and semantic space to perform classification.

There are different kinds of class-level semantic embeddings used by all these algorithms. Some of them are obtained through human annotations [16][32], pre-trained word2vec-model [18], sentence embeddings learned using language model [22], etc.

Few-shot Learning In few-shot learning, very few examples for novel classes are available while training. Directly learning with such small number of examples causes the model to over fit. Earlier approaches like Relation-Net [27] learns deep metric representation to compare the images by simulating few-shot scenario in episode-based mini-batches. Siamese neural network [14] employs CNN architecture to measure the similarity between image pairs for final ranking. Prototypical networks [25], Matching networks [31] use meta-learning approaches to make inferences on the few labeled instances of novel classes. [11] proposes to generate samples of novel classes considering the possible variations, whereas [3] utilizes GAN to transfer the styles captured from base classes to generate data for the novel classes. [21] imprints normalized image embeddings as classifier weights for the novel classes.

In our work, we also aim to address the problem of ZSL, GZSL and FSL by means of a generative model. In contrast to existing work, we propose to generate more meaningful image features by introducing a bidirectional mapping between image and semantic spaces. However, one major disadvantage of all such existing generative models is the need to train an additional classifier while testing. We aim towards addressing such drawbacks. We'll discuss the details of our approach in the following section.

3. Proposed Approach

In this section, we discuss the proposed method for ZSL and GZSL. This method can also be seamlessly extended to the application of few-shot learning, which is elaborated later. For ZSL, we assume the set of seen classes as $\mathcal{C}_{seen} = \{1,..,L\}$ and the set of unseen classes as $\mathcal{C}_{unseen} = \{L+1,..,L+K\}$. As per the standard protocol, $\mathcal{C}_{seen} \cap \mathcal{C}_{unseen} = \phi$.

The training data is given as $\mathcal{S} = \{(\mathbf{x},y)|y \in \mathcal{C}_{seen}\}$, where, $\mathbf{x} \in \mathbb{R}^{d_x}$ denotes the image feature and y denotes the corresponding class label. While testing, given an image $\mathbf{x}_{test} \in \mathbb{R}^{d_x}$, the objective is to learn a mapping function $f_{zsl} : \mathbb{R}^{d_x} \rightarrow \mathcal{C}_{unseen}$ for ZSL, and $f_{gzsl} : \mathbb{R}_{d_x} \rightarrow \mathcal{C}_{seen} \cup \mathcal{C}_{unseen}$ for GZSL. In addition to this, we have the attributes / class-label emeddings for both seen and unseen classes as $c(y) \in \mathbb{R}^{d_a}$, where $y \in \mathcal{C}_{seen} \cup \mathcal{C}_{unseen}$.

Base GAN-based Generative Network Motivated by the success of the generative approaches [35][36] for the task of ZSL, in this work, we propose an improved generative framework for the same. We start the discussion by providing a brief introduction to the base network.

In our work, we use the f-WGAN [35] as the base generative model and propose a better approach with bidirectional exchange of information between visual and semantic domain for feature generation. f-WGAN consists
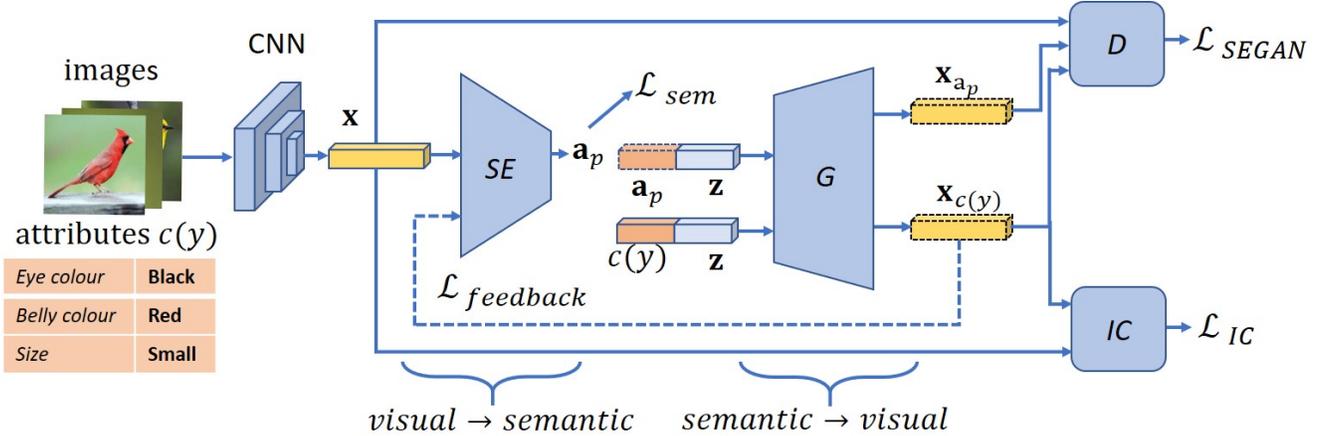
Figure 1. Our generative model utilizes bidirectional mapping, visual to semantic (SE) and semantic to visual (G) to generate the features. Proposed Integrated-classifier (IC) is trained along with the generator (G) and directly used to classify the test instance in contrast to the existing approaches. Retraining of the classifier is not required for our framework.

of two modules - a conditional generator (G) and a conditional discriminator (D), which are trained in an adversarial fashion to capture the underlying distribution of the training data, denoted as P_s. Using the training data of the seen classes $\mathcal{S}$, the parameters of the generator (G) and discriminator (D) are learnt by optimizing the following WGAN loss function [10] conditioned on $c(y)$ as in [35],

$$\mathcal{L}_{WGAN}|_{c(y)} = \mathop{\mathbb{E}}_{\mathbf{x} \sim P_s} [D(\mathbf{x}|c(y))] - \mathop{\mathbb{E}}_{\tilde{\mathbf{x}} \sim P_g} [D(\tilde{\mathbf{x}}|c(y))] \\ - \lambda(GP) \quad (1)$$

Here, $\tilde{\mathbf{x}}$ is the fake image feature generated by G, i.e. $\tilde{\mathbf{x}} = G(\mathbf{z}|c(y))$, where $\mathbf{z}$ is sampled from a pre-defined noise distribution P_z and P_g is assumed to model the generator distribution. GP is the gradient-penalty term, proposed in [10], as

$$GP = \mathop{\mathbb{E}}_{\hat{\mathbf{x}} \sim P_{\hat{x}}} [(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}}|c(y))\|_2 - 1)^2] \quad (2)$$

Similar to [10], $P_{\hat{x}}$ is considered to be uniformly sampled along the straight line between data-pairs sampled from P_s and P_g. Thus, $\hat{\mathbf{x}} = \alpha \mathbf{x} + (1 - \alpha)\tilde{\mathbf{x}}$ [10]. λ is the penalty coefficient, and the suggested value ($\lambda = 10$) [10] is used for our work.

In this work, given this base generative network, we propose two novel modules, namely *Semantic Embedding* and *Integrated Classifier* to improve the quality of the generated image features as well as the final recognition performance. Here, we discuss these modules in detail. Figure 1 depicts a pictorial representation of the proposed framework.

3.1. Semantic Embedding Module

Semantic Embedding (SE) module is designed to capture meaningful semantic inferences from an input image, which can further be utilized in the generation of fake image features using the base f-WGAN. The generation process in the base f-WGAN is conditioned on the attribute information of the class (equation (1)). SE-module explores the possibility of additionally using predicted attributes from the images, which contains some amount of image specific information to generate more diverse image features for each class. In particular, we aim to learn a mapping, $\mathcal{E}_{sem} : \mathbb{R}^{d_x} \rightarrow \mathbb{R}^{d_a}$, and use the predicted semantic embedding $\mathbf{a}_p = \mathcal{E}_{sem}(\mathbf{x})$ as an additional input to f-WGAN.

Given a set of images with their class labels and corresponding attributes, we now describe the trainining of SE to obtain the semantic embeddings. We want the predicted attributes from the images of same class should be as close as possible. In addition, we want these embeddings to be close to the ground-truth attributes ($c(y)$), which incorporate useful class-specific information required for recognition. Motivated by [25], we compute the class-prediction capability of $\mathbf{a}_p$ using the distance-based softmax function as

$$\Pr(\mathbf{y} = k|\mathbf{x}, c(k)) = \frac{\exp(-d(\mathbf{a}_p, c(k)))}{\sum_{k'} \exp(-d(\mathbf{a}_p, c(k')))} \quad (3)$$

where, $d(\mathbf{a}, \mathbf{b})$ denotes the Euclidean distance between two vectors $\mathbf{a}$ and $\mathbf{b}$. We define the negative log-likelihood function on this probability measure as the primary loss component [25] for SE-module as,

$$\mathcal{L}_{pr} = - \mathop{\mathbb{E}}_{\mathbf{x} \sim P_s} [\log \Pr(y = k|\mathbf{x}, c(k))] \quad (4)$$

We further observe experimentally, that mean squared error (MSE) loss helps to achieve better alignment of $\mathbf{a}_p$'s towards their ground-truths $c(y)$. Thus, additionally we impose MSE loss as,

$$\mathcal{L}_{mse} = \mathop{\mathbb{E}}_{\mathbf{x} \sim P_s} [(\mathcal{E}_{sem}(\mathbf{x}) - c(y))^2] \quad (5)$$

Hence, the overall loss function of this SE-module is,

$$\mathcal{L}_{SE} = \mathcal{L}_{pr} + \alpha \mathcal{L}_{mse} \qquad (6)$$

where α is a hyper-parameter, set on the basis of validation set accuracy. In the following section, we will discuss the training methodology of the SE-module jointly with the WGAN-based generative network for generation of better image features. The WGAN with the SE module is termed as SE-GAN in the rest of the paper.

Generating image features with SE-GAN: In the proposed framework, the SE module and the G-module of the WGAN are trained together, in adversarial fashion with D. Only the training set $\mathcal{S}$ is used for this training. The training of G-module in addition to SE is done in an alternate manner in two stages. In the first stage, for a mini-batch, $\mathcal{E}_{sem}$ is trained using $(\mathbf{x}, y) \in \mathcal{S}$ by minimizing the loss $\mathcal{L}_{SE}$. Hence, for the first stage, $\mathcal{L}^{(I)}_{SE-GAN} = \mathcal{L}_{SE}$. In the second stage, for the same mini-batch, $\mathcal{E}_{sem}$ is frozen and the learning of G is initiated.

For training G in the second stage, we modify the standard training of f-WGAN [35] by utilizing both the ground-truth attributes of data $c(y)$, as well as the inferred attributes $\mathbf{a}_p$. Thus, we obtain two sets of fake image data at the generator output, one is obtained using the ground-truth attributes as $\tilde{\mathbf{x}}_{c(y)} = G(\mathbf{z}|c(y))$, and the other one is generated using the predicted attributes as, $\tilde{\mathbf{x}}_a = G(\mathbf{z}|\mathbf{a}_p)$, while the noise component $\mathbf{z} \sim P_z$. To ensure that the distribution of the generated fake image features follows the actual data distribution P_s as closely as possible, both $\tilde{\mathbf{x}}_{c(y)}$ and $\tilde{\mathbf{x}}_a$ are fed to the discriminator in addition to real $\mathbf{x} \in \mathcal{S}$.

Also, we aim to learn features in order to be discriminative across the classes. Towards that goal, we first pre-train a soft-max classifier, G_{cls} on the training set $\mathcal{S}$. While training the SE-GAN, we use this pre-trained classifier to minimize the cross-entropy loss as

$$\mathcal{L}_{cls} = - \underset{\mathbf{x} \sim P_s}{\mathbb{E}} [\log \Pr(y|\tilde{\mathbf{x}}_{c(y)})] - \underset{\mathbf{x} \sim P_s}{\mathbb{E}} [\log \Pr(y|\tilde{\mathbf{x}}_a)] \quad (7)$$

where, both the probablities $\Pr(y|\tilde{\mathbf{x}}_{c(y)})$ and $\Pr(y|\tilde{\mathbf{x}}_a)$ are measured as standard soft-max function. Note that since G_{cls} is pre-trained and kept frozen while training SE-GAN, the back-propagation at this stage will update only the generator module based on the class-discrimination property of the generated features.

Additionally, we propose to feed these fake generated features back to $\mathcal{E}_{sem}$ module to ensure that they can indeed give the class-specific semantic embeddings, which in turn leads to further improvement in the image features. For this stage also, we freeze the weights of $\mathcal{E}_{sem}$ and minimize the loss $\mathcal{L}_{SE}$ computed using $\tilde{\mathbf{x}}_{c(y)}$, which we refer to as $\mathcal{L}^{feedback}_{SE}$. Please note that this minimization will only update the weights in G and thus improve the feature genera-

tion performance. Hence, the SE-GAN loss function for the second stage of training is as follows

$$\mathcal{L}^{(II)}_{SE-GAN} = \mathcal{L}_{WGAN}|_{c(y)} + \beta_1 \mathcal{L}_{WGAN}|_{\mathbf{a}_p} + \beta_2 \mathcal{L}_{cls} + \beta_3 \mathcal{L}^{feedback}_{SE} \qquad (8)$$

Here, $\beta_1, \beta_2, \beta_3$ are hyper-parameters, which are set experimentally. We are introducing bi-directional mapping i.e. visual to semantic via SE module (6) and semantic to visual via the feedback-loss component (8).

While testing, such generated image features from SE-GAN can be used to train a softmax classifier, which can be used for predicting the class of an unseen image sample, as per the standard practice [35][36]. We further propose to eliminate the need for training a separate classifier while testing and integrate the classification process with SE-GAN. We will discuss the process of integrating the proposed classifier in the following section.

3.2. SE-GAN with Integrated Classifier: ISE-GAN

Existing generative-approaches [35][36][19] for ZSL work on the principle of generating samples of unseen classes from their attributes and use such fake samples to train a classifier. The class of the test image sample is then predicted based on this newly trained classifier. Though effective, still this approach has a few significant drawbacks: (1) It requires to generate sufficient number of fake samples (which is an experimental hyper-parameter in the range of 100-2000 [35][36][6]) to be able to train the classifier. (2) In case, a new unseen class is encountered after deploying this model, the classifier needs to be re-trained. All of these, in effect, increase the testing cost of the model. In contrast, we propose to take the classification one step further, by integrating it with the generation, which overcomes both the above limitations. In addition, only few number of generated samples (analysis shown later) are sufficient to achieve a reasonable performance for both ZSL and GZSL.

We name proposed classification module as the Integrated Classifier (IC-module), since it is trained simultaneously with the generator, and thus eliminates the need to train a separate classifier at testing. The IC-module works as a class-discriminative network, which takes a pair of image features as input and outputs a similarity score (scalar value) indicating if the features are from same class or different classes. We generate paired real-fake image feature data, conditioning on the ground-truth attributes of the same for training IC-module as

$$\mathcal{D}_1 = \{(\mathbf{x}^i, \tilde{\mathbf{x}}^j_{c(y)}, c(j)) | i, j \in \mathcal{C}_{seen}\} \qquad (9)$$

Here, we slightly modify the feature notation as $\mathbf{x}^i$ to denote that the feature $\mathbf{x}$ belongs to i^{th} class. Similarly, $\tilde{\mathbf{x}}^j_{c(y)}$ denotes that the fake features $\tilde{\mathbf{x}}_{c(y)}$, generated us-

ing ground-truth attributes belongs to j^{th} class. Additionally, the semantic-embeddings for the unseen classes can be utilized to further generalize the IC-module towards the unseen classes by constructing a fake-fake image feature datasets from $\mathcal{C}_{unseen}$,

$$\mathcal{D}_2 = \{(\tilde{\mathbf{x}}_{c(y)}^i, \tilde{\mathbf{x}}_{c(y)}^j, c(j)) | i, j \in \mathcal{C}_{unseen}\} \quad (10)$$

We can use both $\mathcal{D}_1$ and $\mathcal{D}_2$ to train the IC-module, learning the function $\mathcal{IC} : \mathbb{R}^{d_x} \times \mathbb{R}^{d_x} \times \mathbb{R}^{d_a} \rightarrow (0,1)$, while minimizing the following MSE loss function

$$\mathcal{L}_{IC} = \underset{(\mathbf{x}^i, \tilde{\mathbf{x}}_{c(y)}^j) \sim P_{s,g}}{\mathbb{E}} [(\mathcal{IC}(\mathbf{x}^i, \tilde{\mathbf{x}}_{c(y)}^j, c(j)) - \mathbf{1}_{(i,j)})^2]$$
$$+ \gamma \underset{(\tilde{\mathbf{x}}^i, \tilde{\mathbf{x}}_{c(y)}^j) \sim P_g}{\mathbb{E}} [(\mathcal{IC}(\tilde{\mathbf{x}}_{c(y)}^i, \tilde{\mathbf{x}}_{c(y)}^j, c(j)) - \mathbf{1}_{(i,j)})^2]$$
$$(11)$$

where, $\mathbf{1}_{(i,j)}$ represents the indicator function with value 1, if and only if $i = j$. Otherwise, $\mathbf{1}_{(i,j)} = 0$. $P_{s,g}$ refers to the joint distribution of both real and generated data. γ is again a hyper-parameter set on the basis of validation set accuracy.

Simultaneous training of IC-module with SE-GAN: For ISE-GAN, the IC-module is trained in the second-phase of mini-batch training of SE-GAN, as explained in Section 3.1. To enable the integrated training, the final SE-GAN loss function is modified as below

$$\mathcal{L}_{ISE-GAN} = \mathcal{L}_{SE-GAN}^{(II)} + \mathcal{L}_{IC} \quad (12)$$

Thus, the optimization problem becomes,

$$\min_{\theta_{SE}, \theta_G, \theta_{IC}} \max_{\theta_D} \mathcal{L}_{ISE-GAN} \quad (13)$$

where, θ_{SE}, θ_G and θ_{IC} represent the trainable parameters of the SE, generator and IC modules of SE-GAN respectively. θ_D represents the parameters of discriminator. Therefore, for each mini-batch, both the G and IC-modules are updated simultaneously.

Algorithm 1 summarizes the training process of proposed ISE-GAN. To train the SE-GAN module, we perform the update steps till 14 in Algorithm 1. ISE-GAN differs significantly with other state-of-the-arts generative models, such as f-VAEGAN [36] or f-WGAN [35]. Our main contributions are the SE and IC modules, which can potentially be integrated with any base generative network for improving its performance.

3.3. Testing

While testing, the objective is to classify a test image sample $\mathbf{x}_{test}$ to a set of possible classes $\mathcal{C}_{test}$. For ZSL, $\mathcal{C}_{test} = \mathcal{C}_{unseen}$, while for GZSL, $\mathcal{C}_{test} = \mathcal{C}_{seen} \cup \mathcal{C}_{unseen}$. Towards the goal of unseen image classification, we obtain

---

Algorithm 1 Algorithm for training ISE-GAN

1: **Input**: $\mathcal{S}, \{c(y) | y \in \mathcal{C}_{seen} \cup \mathcal{C}_{unseen}\}$
2: **Initialize**: Randomly initialize the parameters of the discriminator (θ_D), generator (θ_G), SE-module (θ_{SE}) and IC-module (θ_{IC}).
3: **Requirement**: Learning rate β, number of iterations of discriminator per generator iteration $= n_{dis}$, batch size B, P_z is considered as Gaussian, $\mathcal{N}(0,1)$.
4: **while** $\theta_D, \theta_G, \theta_{SE}, \theta_{IC}$ have not converged, **do**
5:     **for** $i = 1, 2, ..., n_{dis}$ **do**
6:         Sample $\{\mathbf{x}_i, c(y_i)\}_{i=1}^B, \{\mathbf{z}_i\}_{i=1}^B \sim P_z$.
7:         $\mathbf{a}_{p_i} = \mathcal{E}_{sem}(\mathbf{x}_i)$
8:         $\mathcal{L}_D(\mathbf{x}_i) = \mathcal{L}_{WGAN} | c(y_i) + \mathcal{L}_{WGAN} | \mathbf{a}_{p_i}$
9:         Gradient: $g_{\theta_D} \leftarrow \nabla_{\theta_D} \frac{1}{B} \sum_{i=1}^B L_D(\mathbf{x}_i)$
10:        Update: $\theta_D \leftarrow \theta_D - \beta \text{Adam}(\theta_D, g_{\theta_D})$
11:        Gradient : $g_{\theta_{SE}} \leftarrow \nabla_{\theta_{SE}} \frac{1}{B} \sum_{i=1}^B \mathcal{L}_{SE-GAN}^{(I)}(\mathbf{x}_i)$
12:        Update: $\theta_{SE} \leftarrow \theta_{SE} - \beta \text{Adam}(\theta_{SE}, g_{\theta_{SE}})$
13:        Gradient: $g_{\theta_G} \leftarrow \nabla_{\theta_G} \frac{1}{B} \sum_{i=1}^B \mathcal{L}_{SE-GAN}^{(II)}(\mathbf{x}_i)|_{GP=0}$
14:        Update: $\theta_G \leftarrow \theta_G - \beta \text{Adam}(\theta_G, g_{\theta_G})$
15:        Construct $\mathcal{D}_1$ and $\mathcal{D}_2$ as equation (9) and (10)
16:        Gradient : $g_{\theta_{IC}} \leftarrow \nabla_{\theta_{IC}} \mathcal{L}_{IC}$
17:        Update: $\theta_{IC} \leftarrow \theta_{IC} - \beta \text{Adam}(\theta_{IC}, g_{\theta_{IC}})$
---

n_g number of fake image samples of each class $y \in \mathcal{C}_{test}$ from its ground-truth attribute $c(y)$ using the trained generator module. Next, we obtain one class prototype for each class as the sample mean given below

$$\mathbf{m}(y) = \frac{1}{n_g} \sum_{k=1}^{n_g} G(\mathbf{z}_k | c(y)) \quad (14)$$

Here, $\mathbf{z}_k$, $k = 1, ..., n_g$ are uniformly sampled from the same training noise distribution P_z. Finally, the test sample $\mathbf{x}_{test}$ is paired with each of the class prototypes and their corresponding class attributes as $(\mathbf{x}_{test}, \mathbf{m}(y), c(y))$ and is fed to the IC-module to generate the class similarity score. The class with the highest similarty score is predicted as the target class (y_{target}) of that sample. Thus,

$$y_{target} = \arg \max_{y \in \mathcal{C}_{test}} \mathcal{IC}(\mathbf{x}_{test}, \mathbf{m}(y), c(y)) \quad (15)$$

This IC-module eliminates the requirement of test-time training of the classifier. Also, this target-class evaluation methodology can be seamlessly extended to any number of classes while testing.

Extension to Few-shot learning: The training procedure of proposed model for FSL remains the same, However, the testing methodology is modified slightly in

case of FSL to accommodate the existing few training samples of novel classes. Utilizing the given training examples, the class prototypes for the novel classes are computed as

$$\mathbf{m}_{fs}(y) = \frac{\mathbf{m}(y) + \sum_{i=1}^{n_{fs}} \mathbf{x}_i}{1 + n_{fs}} \qquad (16)$$

where n_{fs} is the number of examples available for training for class y. Following this, the prediction of y_{target} is exactly same as in the case of ZSL.

Dataset	d_a	Total classes	$|\mathcal{C}_{seen}|$	$|\mathcal{C}_{unseen}|$
CUB	312	200	100 + 50	50
SUN	102	717	580 + 65	72
AWA1	85	50	27 + 13	10
AWA2	85	50	27 + 13	10

Table 1. Datasets details in terms of dimension of attributes (d_a), number of total classes and the split in terms of seen/unseen classes. $|\mathcal{C}_{seen}|$ column contains number of seen classes in the format no. of training classes + no. of validation classes.

4. Experiments

In this section, we discuss the details of experiments performed to evaluate the effectiveness of the proposed framework. We begin with a brief description of the datasets and the features used for the experiments.

4.1. Datasets and Features Used

We evaluate the proposed approach on four widely used benchmark datasets, namely Caltech-UCSD Birds (CUB)-200-2011 [32], SUN [20], Animals with Attributes1 (AWA1) [16] and Animals with Attributes2 (AWA2) [34] for both ZSL and GZSL settings. For all our experiments, we use the features (2048-d) extracted from final pooling layer of ResNet-101 [12], pre-trained on ImageNet [5], to represent the images. The manual attribute annotations of images in all four datasets are used as the semantic embedding to generate the results in all cases, unless specified otherwise.
Caltech-UCSD Birds 200-2011 (CUB) [32] is a fine-grained dataset, with 11,788 images of 200 different birds, annotated with 312-d manual attributes.
SUN [20] is another fine-grained dataset, with 14,340 scene images from 717 classes annotated with 102-d attributes.
Animals with Attributes1 (AWA1) [16] has 30,475 images from 50 classes, annotated with 85 attributes.
Animals with Attributes2 (AWA2) [34] is a newly released dataset with 37,322 images (non-overlapping with AWA1) from 50 classes annotated with 85 attributes.

The data-splits for ZSL are done following [34]. The summary of the split is depicted in Table 1.

4.2. Implementation Details

In our implementation, the generator and discriminator are multilayer perceptron (MLP) with two hidden layers and 4096 hidden units. The IC and SE-modules are implemented as single hidden layer MLP with 4096 nodes. All the hidden nodes are activated with LeakyRelu except for final layer in the generator, where ReLU-activation is used to match with the ResNet101 final layer features. We use a batch size of 64, Adam optimizer with a learning rate of 0.0001. We explicitly mention all hyper-parameter values in Table 3 for ease of replicating the results.

Dataset	α	β_1	β_2	β_3	γ
AWA1 & AWA2	0.2	1	0.1	0.2	0.5
CUB	0.05	0.01	0.01	0.1	0.75
SUN	0.1	0.05	0.01	0.1	0.25

Table 3. Hyper-parameter values used in our work.

4.3. Generalized Zero-Shot Learning

Here, we report the experiments for GZSL protocol as used in [34] for the proposed framework. To demonstrate the effectiveness of the proposed approach, we compare with the recent state-of-the-art methods :

- **Non-deep traditional methods :** CMT [26], SJE [2], ALE [1], LATEM [33], ESZSL [23], SYNC [4];

- **Deep learning based methods:** DeViSE [7], f-CLSWGAN [35], SE [29], ReViSE [28], f-VAEGAN [36], CADA-VAE [24] etc.

The summary of the experiments on all the four datatsets are tabulated in Table 2. The results for all the other approaches are reported directly from [36] and [24].

We perform all the experiments with two variants of our proposed framework. First, we use SE-GAN module to generate high-quality synthetic image features for unseen classes and then train a softmax classifier for final classification. This variation of the model is referred to as 'SE-GAN+SM' in Table 2. On the other hand, the complete framework has been denoted as 'ISE-GAN'. We observe that for all four datasets, we are able to achieve highest Harmonic-mean value compared to the state-of-the-art CADA-VAE [24]. We also observe that in general, the ISE-GAN with the integrated classifier module performs better than the other version (SE-GAN+SM), signifying the effectiveness of the integrated classifier in generating more useful and discriminative image features. To further evaluate the effectiveness of the proposed approach, we also perform experiments for the ZSL settings. Here we follow the same experimental protocol as [34] and use the same features. As before, we experimented with both the variants of the proposed framework. The results in terms of top-1% accuracy

Model	CUB			SUN			AWA1			AWA2		
	S	U	H	S	U	H	S	U	H	S	U	H
CMT [26]	49.8	7.2	12.6	21.8	8.1	11.8	87.6	0.9	1.8	90.0	0.5	1.0
SJE [2]	59.2	23.5	33.6	30.5	14.7	19.8	74.6	11.3	19.6	73.9	8.0	14.4
ALE [1]	62.8	23.7	34.4	33.1	21.8	26.3	76.1	16.8	27.5	81.8	14.0	23.9
LATEM [33]	57.3	15.2	24.0	28.8	14.7	19.5	71.7	7.3	13.3	77.3	11.5	20.0
ESZSL [23]	63.8	12.6	21.0	27.9	11.0	15.8	75.6	6.6	12.1	77.8	5.9	11.0
SYNC [4]	70.9	11.5	19.8	43.3	7.9	13.4	87.3	8.9	16.2	90.5	10.0	18.0
DeViSE [7]	53.0	23.8	32.8	27.4	16.9	20.9	68.7	13.4	22.4	74.7	17.1	27.8
CDL [13]	55.2	23.5	32.9	34.7	21.5	26.5	73.5	28.1	40.6	-	-	-
f-CLSWGAN [35]	57.7	43.7	49.7	36.6	42.6	39.4	61.4	57.9	59.6	68.9	52.1	59.4
SE [29]	53.3	41.5	46.7	30.5	40.9	34.9	67.8	56.3	61.5	68.1	58.3	62.8
ReViSE [28]	28.3	37.6	32.3	20.1	24.3	22.0	37.1	46.1	41.1	39.7	46.4	42.8
f-VAEGAN [36]	60.1	48.4	53.6	38.0	45.1	41.3	-	-	-	70.6	57.6	63.5
CADA-VAE [24]	53.5	51.6	52.4	35.7	47.2	40.6	72.8	57.3	64.1	75.0	55.8	63.9
SE-GAN+SM	57.6	48.4	52.6	37.0	44.7	40.5	68.3	53.9	60.3	61.9	55.1	58.3
ISE-GAN	55.4	52.4	**53.8**	34.7	51.3	**41.4**	74.4	58.7	**65.6**	79.3	55.9	**65.5**

Table 2. Evaluation of SE-GAN+Soft-max (SM) classifier and ISE-GAN using average per-class top-1 accuracy (%) for GZSL protocol. The results have been compared with the state-of-the-art. "H" denotes the harmonic mean of the classification performance evaluated using the accuracy for unseen ("U") and seen ("S") classes.

and comparisons with state-of-the-art approaches are summarized in Table 4. All the results for the other approaches are directly reported from [24]. We observe that the proposed framework gives state-of-the-art results for ZSL task for three out of the four datasets and gives comparable performance for AWA2 dataset.

4.4. Few-shot Learning

We further conduct experiments for few-shot learning (FSL) setting on CUB and SUN datasets. We compare the results for the same with the most recent relevant work reported in [24]. Let us denote the number of samples available for the novel classes as n_{fs}. For comparison, we use all the unseen classes ($\mathcal{C}_{unseen}$) with $n_{fs} = 0, 2, 5$ and 10, thus evaluating the algorithm for $\mathcal{C}_{unseen}$-way n_{fs}-shot protocol in FSL setting. The results and comparison with state-of-the-art are shown in Figure 2. All the results for FSL are generated using our full proposed framework ISE-GAN. We observe that on CUB dataset, the proposed method ISE-

GAN outperforms both the algorithms with noticable margin. On SUN dataset, ReViSE [28] performs slightly better compared to the proposed method for $n_{fs} = 0$. However, the accuracy of the proposed method improves significantly over both [28] and [24] even with $n_{fs} = 2$.

Model	CUB	SUN	AWA1	AWA2
CMT [26]	34.6	39.9	39.5	37.9
SJE [2]	53.9	53.7	65.6	61.9
ALE [1]	54.9	58.1	59.9	62.5
LATEM [33]	49.3	55.3	55.1	55.8
ESZSL [23]	53.9	54.5	58.2	58.6
SYNC [4]	55.6	56.3	54.0	46.6
CDL [13]	54.5	63.6	69.9	-
DeViSE [7]	52.0	56.5	54.2	59.7
f-CLSWGAN [35]	57.3	60.8	68.2	-
SE [29]	59.6	63.4	69.5	69.2
f-VAEGAN [36]	61.0	**64.7**	-	**71.1**
CADA-VAE [24]	60.4	61.8	62.3	64.0
SE-GAN+SM	60.8	61.8	**70.2**	68.8
ISE-GAN	**63.9**	**64.7**	68.4	65.6

Table 4. Evaluation of SE-GAN+SM-classifier and ISE-GAN using average per-class top-1 accuracy (%) and comparison with the state-of-the-art for ZSL protocol.

Figure 2. Comparing the performance of ISE-GAN with state-of-the-art CADA-VAE [24] and ReViSE [28] with respect to increasing number of training examples for few-shot learning.

4.5. Analysis

In this section, we perform detailed analysis of the proposed framework. We report the results of all the analysis

on the CUB dataset, unless specified otherwise.

Effect of number of synthetic features generated: In ISE-GAN, the synthetic image features generated are used to compute the prototype ($\mathbf{m}(y)$) for each class, which are finally used to determine the class of test image. Thus the final performance depends on the quality of the prototypes, which in turn depends on the quality and quantity of the synthetic features. We analyze the performance of ISE-GAN with varying number of synthetic features used to compute the prototype. The classification accuracy in terms of top-1% accuracy (for ZSL) and Harmonic mean (for GZSL) for fine-grained CUB and coarse-grained AWA1 datasets are shown in Figure 3. We observe that the number of generated
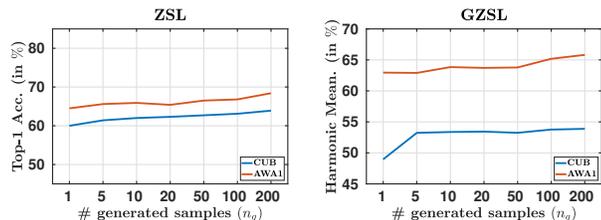
Figure 3. Number of samples required (n_g) to generate mean vector ($m(y)$) vs. classification performance of ISE-GAN.

features required for the proposed ISE-GAN is very less compared to the separate softmax-classifier. For instance, the number of samples to be generated for SE-GAN with softmax-classifier (SE-GAN+SM) is ~ 2000 per class for AWA1 under GZSL protocol, whereas using ISE-GAN, it is 200. Using ISE-GAN, even with a single generated feature, on AWA1 dataset, we obtain an accuracy of 64.5% for ZSL and 62.9% for GZSL.

Effect of different class embeddings: Here, we explore the effect of different attribute representations on the proposed framework. In CUB dataset, in addition to the manual attributes, each image is also annotated with 10-sentence long textual description. Figure 4 reports the results using sentence-embeddings (stc) [22] of the textual description and Word2Vec (w2v) [18] embeddings extracted using class names. We observe that ISE-GAN performs better using
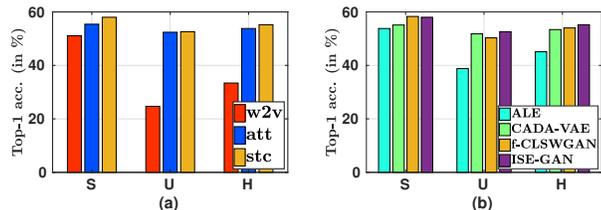
Figure 4. For GZSL protocol on CUB dataset: (a) Effect of different semantic embeddings on the classification performance of ISE-GAN; (b): With sentence-embeddings as $c(y)$, performance comparison of ISE-GAN with recent state-of-the-art.

stc-embeddings as compared to the other representations.

ISE-GAN outperforms all the other state-of-the-arts irrespective of class embeddings, justifying the robustness of our method under different class embeddings.

Ablation Study: Here, we analyze the behavior of proposed model subject to different variation of the proposed loss function and the results are summarized in Table 5. First, we train our model (**V1**) using the synthetic features generated with only the ground-truth attributes ($c(y)$) and the classification loss component ($\mathcal{L}_{SE-GAN}(\beta_1 = 0, \beta_3 = 0)$). This essentially makes the model similar to [35]. Next, we incorporate the proposed SE module with [35], but remove the loss components $\mathcal{L}_{cls}$ and $\mathcal{L}_{SE}^{feedback}$ ($\beta_2 = 0, \beta_3 = 0$ in equation (8)). We refer to this part of the framework as **V2**. We observe that the results improve over [35]. Next, we add only $\mathcal{L}_{SE}^{feedback}$ to **V2** to obtain **V3** and observe further boost in performance, specially for GZSL. This reflects the effectiveness of feedback loss component in capturing the coherency between semantic space and generated features. To observe the effect of $\mathcal{L}_{cls}$, we remove $\mathcal{L}_{SE}^{feedback}$ from **V3** and train the model (**V4**). We observe an improvement of performance, which validates our idea of discriminative feature generation. Finally, the full framework ISE-GAN yields the best performance over all of the above variations with both softmax classifier as well as IC-module.

Model	Classifier	GZSL	ZSL
V1: f-CLSWGAN [35]	SM	49.7	57.3
V2: $\mathcal{L}_{SE-GAN}$ ($\beta_2 = 0, \beta_3 = 0$)	SM	50.9	58.5
V3: $\mathcal{L}_{SE-GAN}$ ($\beta_2 = 0$)	SM	51.4	58.8
V4: $\mathcal{L}_{SE-GAN}$ ($\beta_3 = 0$)	SM	51.9	59.5
$\mathcal{L}_{SE-GAN}$	SM	52.6	60.8
$\mathcal{L}_{SE-GAN}$	IC	**53.8**	**63.9**

Table 5. Ablation study of the proposed framework on CUB dataset for both ZSL (Top-1 acc (%)) and GZSL (Harmonic mean of "S" and "U" (%)).

5. Conclusion

In this work, we proposed two novel deep network-modules to enhance the performance of generative approaches for the task of ZSL, GZSL and FSL. SE module introduces a bi-directional mapping between the semantic space and the image feature space to boost the generation performance of f-WGAN. In comparison to the existing model used for the same, our method has the advantage of an integrated classifier, which eliminates the requirement of training of a separate classifier, while testing. Our extensive experimental evaluation on four datasets shows that the proposed framework either outperforms or performs comparably to the state-of-the-art. A detailed ablation study of the model shows the effectiveness of each module and loss component in the network.

References

[1] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7):1425–1438, 2016. 1, 2, 6, 7

[2] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 1, 2, 6, 7

[3] A. Antoniou, A. Storkey, and H. Edwards. Data augmentation generative adversarial networks. In *International Conference on Learning Representations Workshops*, 2018. 2

[4] S. Changpinyo, W. Chao, B. Gong, and F. Sha. Synthesized classifiers for zero-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1, 6, 7

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Computer Vision and Pattern Recognition*, 2009. 6

[6] R. Felix, B. G. V. Kumar, I. D. Reid, and G. Carneiro. Multimodal cycle-consistent generalized zero-shot learning. *European Conference on Computer Vision*, 2018. 1, 2, 4

[7] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. Devise: a deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems*, 2013. 2, 6, 7

[8] S. Gidaris and N. Komodakis. Dynamic few-shot visual learning without forgetting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014. 2

[10] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, 2017. 2, 3

[11] B. Hariharan and R. Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *IEEE International Conference on Computer Vision*, 2017. 1, 2

[12] K. He, S. Zhang, and J. Sun. Deep residual learning for image recognition. In *IEEE International Conference on Computer Vision*, 2015. 6

[13] H. Jiang, R. Wang, S. Shan, and X. Chen. Learning class prototypes via structure alignment for zero-shot recognition. In *European conference on computer vision (ECCV)*, 2018. 2, 7

[14] G. Koch and R. Zemel. Siamese neural networks for one-shot image recognition. In *International Conference on Machine Learning*, 2015. 2

[15] E. Kodirov, T. Xiang, and S. Gong. Semantic autoencoder for zero-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2

[16] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2014. 1, 2, 6

[17] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. In *International Conference on Machine Learning*, 2016. 2

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 2013. 2, 8

[19] A. Mishra, S. K. Reddy, A. Mittal, and H. A. Murthy. A generative model for zero shot learning using conditional variational autoencoders. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2018. 1, 2, 4

[20] G. Patterson and J. Hays. Sun attribute database: discovering, annotating, and recognizing scene attributes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 6

[21] H. Qi, M. Brown, and D. G. Lowe. Low-shot learning with imprinted weights. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[22] S. Reed, Z. Akata, H. Lee, and B. Schiele. Learning deep representations of fine-grained visual descriptions. In *IEEE Computer Vision and Pattern Recognition*, 2016. 2, 8

[23] B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. In *International Conference on Machine Learning*, 2015. 1, 6, 7

[24] E. Schonfeld, S. Ebrahimi, S. Sinha, T. Darrell, and Z. Akata. Generalized zero-shot learning via aligned variational autoencoders. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 6, 7

[25] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 2017. 1, 2, 3

[26] R. Socher, M. Ganjoo, C. D. Manning, and A. Y. Ng. Zero-shot learning through cross-modal transfer. In *Advances in Neural Information Processing Systems*, 2013. 2, 6, 7

[27] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales. Learning to compare: relation network for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[28] T. Y.-H. Tsai, L. K. Huang, and R. Salakhutdinov. Learning robust visual-semantic embeddings. In *IEEE International Conference on Computer Vision*, 2017. 2, 6, 7

[29] V. K. Verma, G. Arora, A. Mishra, and P. Rai. Generalized zero-shot learning via synthesized examples. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2, 6, 7

[30] V. K. Verma and P. Rai. A simple exponential family framework for zero-shot learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2017. 2

[31] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, and et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, 2016. 1, 2

[32] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. *Computation and Neural Systems Technical Report, CNS-TR-2011-001*, 2011. 2, 6

[33] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele. Latent embeddings for zero-shot classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2, 6, 7

[34] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2251–2265, 2018. 1, 6

[35] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata. Feature generating networks for zero-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2, 3, 4, 5, 6, 7, 8

[36] Y. Xian, S. Sharma, B. Schiele, and Z. Akata. f-VAEGAN-D2: a feature generating framework for any-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 4, 5, 6, 7