CANZSL: Cycle-Consistent Adversarial Networks for Zero-Shot Learning from Natural Language

Zhi Chen
School of ITEE,
University of Queensland
uqzhichen@gmail.com

Jingjing Li
University of Electronic Science
and Technology of China
lijin117@yeah.net

Yadan Luo
School of ITEE,
University of Queensland
lyadanluol@gmail.com

Zi Huang
School of ITEE,
University of Queensland
huang@itee.uq.edu.au

Yangyang
University of Electronic
Science and Technology of China
dlyyang@gmail.com

Abstract

Existing methods using generative adversarial approaches for Zero-Shot Learning (ZSL) aim to generate realistic visual features from class semantics by a single generative network, which is highly under-constrained. As a result, the previous methods cannot guarantee that the generated visual features can truthfully reflect the corresponding semantics. To address this issue, we propose a novel method named Cycle-consistent Adversarial Networks for Zero-Shot Learning (CANZSL). It encourages a visual feature generator to synthesize realistic visual features from semantics, and then inversely translate back synthesized the visual feature to corresponding semantic space by a semantic feature generator. Furthermore, in this paper a more challenging and practical ZSL problem is considered where the original semantics are from natural language with irrelevant words instead of clean semantics that are widely used in previous work. Specifically, a multi-modal consistent bidirectional generative adversarial network is trained to handle unseen instances by leveraging noise in the natural language. A forward one-to-many mapping from one text description to multiple visual features is coupled with an inverse many-to-one mapping from the visual space to the semantic space. Thus, a multi-modal cycle-consistency loss between the synthesized semantic representations and the ground truth can be learned and leveraged to enforce the generated semantic features to approximate to the real distribution in semantic space. Extensive experiments are conducted to demonstrate that our method consistently outperforms state-of-the-art approaches on natural language-based zero-shot learning tasks.

1. Introduction

Over the past few years, deep learning techniques have remarkably boosted the performance of object classification tasks. This success is attributed to the availability of enormous amount of data for training. However, it is unlikely to collect training data for every class in the real world. In order to tackle such a problematic situation, zero-shot learning [38] as a promising solution to recognize new categories with limited training categories has been widely researched recently. The early ZSL aims to find an intermediate semantic representation to transfer the knowledge learned from seen categories to unseen ones. Recently, generative methods are further studied to directly synthesize unseen visual features.

Existing generative approaches synthesizing visual features for unseen classes with Generative Adversarial Networks (GANs) [43, 6, 36] are proposed to address the data missing problem of unseen classes. Generally, the class semantic prototypes together with some noises are fed into these generative models to enforce the synthesized visual features as realistic as the real visual features. Once the models are trained, plausible visual features can be generated given semantic prototypes of unseen classes. However, merely based on adversarial losses, the visual feature generators cannot guarantee that the synthesized features truthfully reflect the corresponding semantics.

On the other hand, existing ZSL methods generally assume that the semantic representations are available and clean. However, it requires domain experts to manually annotate attributes. Furthermore, collecting hundreds of attributes as semantic representations for each category is extremely time-consuming and tedious. In contrast, online noisy text descriptions, e.g., Wikipedia articles, are much

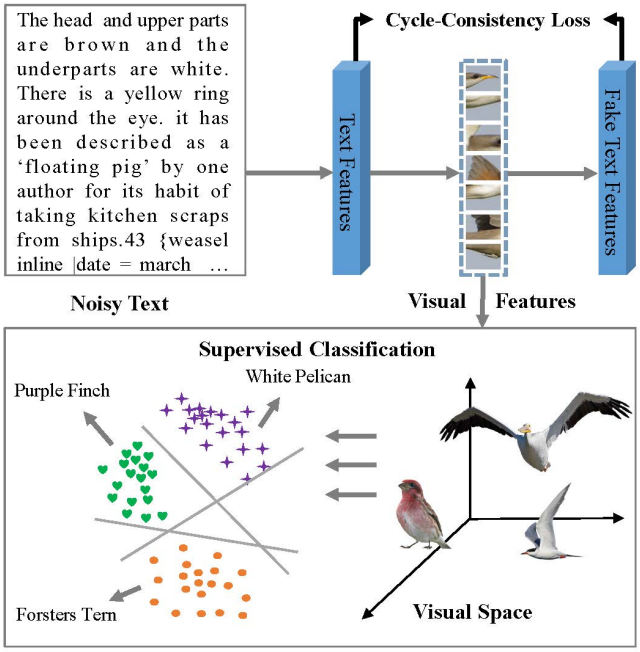

Figure 1. Our CANZSL model leverages cycle-consistent adversarial networks to synthesize realistic visual features from natural language. The problem is simplified as training a supervised classifier to predict the image labels.

easier to collect. And, more excitingly, they are free.

To address these two issues, we propose a novel cycle model, as shown in Fig. 1, to synthesize realistic and discriminative visual features from noisy text representations. A supervised classifier can be simply trained to predict the labels of unseen objects with the synthesized visual features, which are sampled from unseen semantic representations. Specifically, there are three main components in our architecture as shown in Fig. 2. Firstly, a fully connected layer is used to denoise and embed the natural language into pure textural representations. Secondly, a WGAN [3] is utilized to leverage the Wasserstein distance between the real and generated visual feature distributions, considering its demonstrated capability of extinguishing mode collapse. Acting as the main component in our framework for visual feature generating, the WGAN is able to generate diverse visual features. In addition to the weight clipping on the discriminator for satisfying the Lipschitz constraint, we further train a classification network on the discriminator. The classifier is adopted to categorize both the synthesized and the real visual features into correct classes. The classification loss also regularizes the generated visual features to be as much discriminative as the real visual features. Thirdly, in order to guarantee that the synthesized visual features can accurately reflect the corresponding semantics, we adopt an inverse adversarial network to convert the synthesized visual features back to textual features. By applying the cycle-consistent loss between the output text features from the inverse GAN and the input text features to the forward GAN, the inverse GAN collaboratively boosts the forward GAN to capture the underlying data structure. Besides, in the inverse discriminator we train a classifier with the same manner of the visual feature classification. We argue that the textual feature classification loss prevents the embedding FC layer from losing semantic information while suppressing the noise.

The main contributions of this paper can be summarized as follows:

1) We propose a novel structure called cycle-consistent adversarial networks for zero-shot learning, which is capable of synthesizing missing visual features for unseen classes from noisy Wikipedia articles.

2) Different from existing approaches, which only deploy a single GAN to learn semantic to visual mapping, our model consists of two symmetric GANs, a forward GAN and an inverse GAN. They collaboratively promote each other by the constraint of the cycle-consistency loss, the adversarial loss, and the classification loss.

3) The performance of the proposed CANZSL is verified on two tasks: zero-shot recognition and generalized zero-shot learning. Extensive experiments on two benchmark datasets, CUB and NAB, demonstrate that the proposed method consistently outperforms state-of-the-art methods.

2. Related Work

2.1. Zero-shot Learning

Zero-shot learning aims to overcome the issue of increasing difficulty in collecting data for a large amount of categories. Most exciting methods for zero-shot learning are attribute-based visual recognition [20, 30, 26, 18] where the object attributes work as an intermediate feature space that transfer knowledge across object categories.

However, unlike well-specified attribute representations, the real world data is mostly natural language, *e.g.*, Wikipedia articles. In this case, there is another research direction that explores zero-shot learning using online text articles. Elhoseiny *et al*. [10] proposed an approach based on regression and domain adaptation that utilizes unpaired textual descriptions and images. Bo *et al*. [21] took advantage of deep convolutional neural network architecture and utilized latent features from different layers, resulting in a remarkable improvement on zero-shot recognition. Qiao *et al*. [28] proposed a noise suppression technique for noisy signal in text based on $l_{2,1} - norm$ and learn a function to match the text document and the visual features. Further, they also analyzed in-depth that which particular information in documents is useful for zero-shot learning.

Another strategy for zero-shot learning converts the zero-shot problem to a traditional supervised classification

task by sampling realistic visual features for unseen categories [14, 15, 43]. Guo *et al*. [14] estimated the probability distribution of unseen classes from the knowledge acquired from seen classes, and trained supervised classifiers according to the samples that are synthesized based on the distribution. They [15] also proposed an approach to synthesize images directly from seen class probability distribution where the noise from images undoubtedly cause side effects. GAZSL [43] adopted a single GAN model to synthesize visual features from semantics and achieved state-of-the-art performance.

2.2. Generative Adversarial Networks

Generative Adversarial Networks (GANs) [12, 39] have demonstrated favorable performance on image generation [37, 8], image editing [41], and representation learning [29, 31]. A GAN consists of a generator and a discriminator, and the idea behind is training the generator that can fool the discriminator to confuse the distributions of the generated and true samples. Theoretically, the training procedure can allow the generator to perfectly model the data distribution. However, it is usually hard to train a GAN, and mode collapse is known as a common issue of GANs due to the lack of explicit constraint in the learning objective. There are many methods [39, 24, 13, 3] recently proposed to stabilize the training procedure of GANs and mitigating model collapse issues, via using alternative objective functions. WGAN [3] introduced the Wasserstein distance among distributions as the objective function, and showed its capability of mitigating mode collapse. They applied weight clipping on the discriminator to allow the Lipschitz constraint. The improved WGAN [13] used an additional gradient penalty instead of weight clipping to get rid of side effects in WGAN. We adopt WGAN with conditional information as the generative model in our proposed architecture and further use the gradient penalty technique to accelerate the convergence of WGAN.

In a conditional setting, compared with a standard GAN, conditional GANs (cGANs) [25, 7] take additional information vectors (e.g., textual descriptions) as input to both generator and discriminator. The additional information enables the generator to synthesize samples corresponding to the given condition. Auxiliary Classifier GAN (ACGAN) [27] adopted extra classification information in the discriminator, which encourages the generator to synthesize samples based on the class labels as well. In the proposed model, the forward GAN and the inverse GAN take text features and visual representations as input, respectively. Also, we train two classifiers in each discriminator so that the classification information can be preserved.

2.3. Cycle Architecture

A cycle consistency error is proposed in addition to adversarial losses by Zhu *et al*. [42] to tackle the problem of lacking image-to-image translation training data. It is worth mentioning that the CycleGAN can be viewed as training two autoencoders [16]; each has a opposite internal structure to the other. Such a setup can also be seen as a special case of "adversarial autoencoders" [23], which trains the bottleneck layer of an autoencoder by using an adversarial loss to approximate an arbitrary target distribution.

The CycleGAN [42] framework is then widely adopted by a number of works. Cycada [17] utilized the cycle model to perform various applications, e.g., digit adaption, cross-sense adaption. CamStyle [40] is a camera style adaption approach proposed to conduct person re-identification tasks based on CycleGAN. It is also applied to cross-model retrieval [35], where a number of hash functions are learned to enable translation between modalities while using the cycle-consistency loss to enforce the correlation between outputs and original inputs. The above mentioned methods demonstrated the superiority of the cycle architecture in various tasks. Inspired by this observation, we apply the cycle architecture in zero-shot learning from natural language.

3. Approach

We first introduce the problem formulation and then discuss in detail the proposed cycle-consistent adversarial networks for ZSL. Lastly, we illustrate our training procedure and how we conduct our zero-shot recognition task.

3.1. Problem Formulation

Given a batch of seen instances defined by N^s triplets $\{(x_i, \alpha_i, y_i)\}_{i=1}^{N^s}$, where $x_i \in \chi^s$ denotes the image features, α_i^s and y_i^s represent the corresponding TF-IDF vector from Wikipedia articles and the associated one-hot class label respectively. Note that the seen instances S and the unseen instances U are disjointed $S \cap U = \varnothing$. In the test phase, it is assumed that the visual feature x_i^u and TF-IDF vector α_i^u of a new category are provided, ZSL aims to predict the category label y_i^u.

3.2. Model Architecture

Our model mainly comprises two components: a forward visual feature synthesis network $F = \{G_1, D_1\}$ and an inverse text feature generation network $V = \{G_2, D_2\}$.

3.2.1 Visual Feature Synthesis Network

Our Forward Network $F = \{G_1, D_1\}$ for visual feature synthesis is a conditional WGAN with auxiliary information. Specifically, it comprises a generator G_1 and a discriminator D_1. We simply use a fully connected layer for text embedding in the generator and regularize the visual features with the mean value in each classes, so that the distances between categories can be preserved in visual space.

Visual Feature Generator G_1: Given the Term Frequency-Inverse Document Frequency (TF-IDF) features

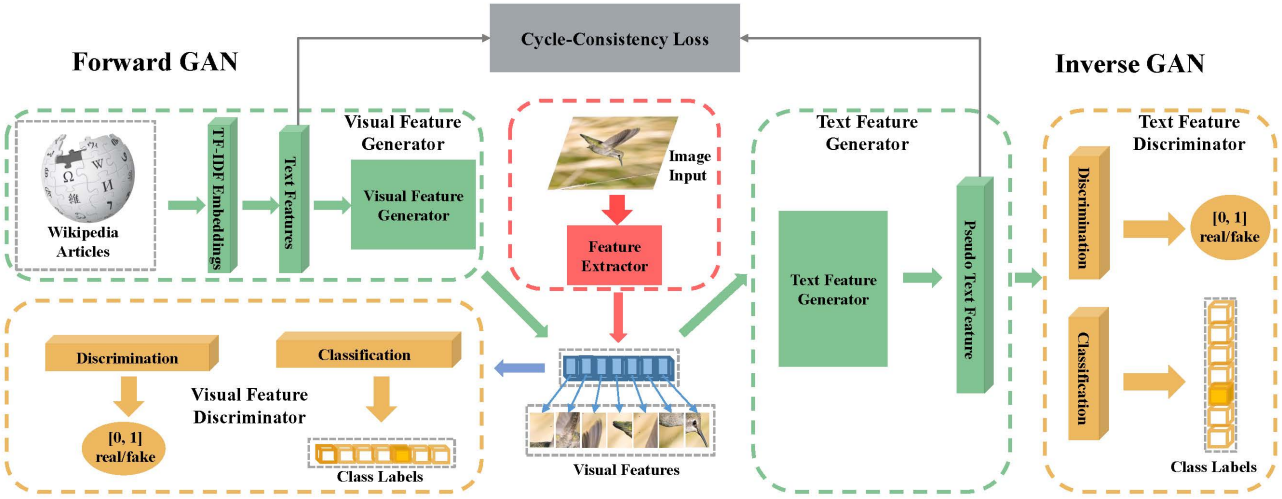

Figure 2. Model overview. Given input images, our approach first extracts deep visual features as real samples throughout the training procedure (the pink part). The forward GAN (on the left hand side) synthesizes realistic and discriminative visual features from the TF-IDF embeddings of Wikipedia articles, whereas the Inverse GAN takes as input the generated visual features to reconstruct into text features again. Then the cycle-consistent loss can be applied to regularize the forward GAN to uncover semantics from the noisy TF-IDF features.

α from the seen natural language descriptions, we first use a fully connected layer as the text encoder $\psi$to generate text embedding $a \leftarrow \psi(\alpha)$. The knowledge-distilled text embedding is then concatenated with a random noise distribution $z \in \mathbb{R}^z$ sampled from Gaussian distribution $N(0, 1)$. The next training step is feeding the concatenated vector $[a, z]$ into two fully connected layers, and each followed by activation functions - Leaky ReLU and Tanh respectively. So far, the fake visual features and the text features $[\hat{x},\ s]$ = $G_1(\alpha, z)$ are generated and the objective of the feature generation network can be formulated as:

$$\begin{aligned}\mathcal{L}_{G_1} = -\mathbb{E}_{z\sim p_z}[D_1(G_1(\alpha, z, \theta), w)] \\ +\mathcal{L}_{cls_1}(G_1(\alpha, z, \theta)),\end{aligned} \tag{1}$$

where the D_1 represents Wasserstein loss [3] and the $\mathcal{L}_{cls_1}$ is the visual feature classification loss according to category labels, which will be introduced in detail in the Discriminator. θ and w signify the parameters in the generator and the discriminator.

Visual Feature Discriminator D_1: The synthesized visual features from visual feature generator G_1 and the real image features are fed into D_1. After the input visual features passing through a fully connected layer and activation function ReLU, we use a fully connected layer to distinguish if the input features are real or not, and simultaneously use another fully connected layer to classify the input image features into correct categories. Introducing classification loss in discriminator has shown its promising effects in Auxiliary Classifier GAN [27]. The objective function of the visual feature discrimination network can be defined as:

$$\begin{aligned}\mathcal{L}_{D_1} = \frac{1}{2}(\mathcal{L}_{cls_1}(G_1(\alpha, z, \theta)) + \mathcal{L}_{cls_1}(x)) + \mathcal{L}_{GP_1} \\ +\mathbb{E}_{z\sim p_z}[D_1(G_1(\alpha, z, \theta), w)] - \mathbb{E}_{x\sim p_{data}}[D_1(x, w)],\end{aligned} \tag{2}$$

where the first two terms are visual feature classification losses, we experimentally set the coefficient for fake features as $\frac{1}{2}$ since it works stably over different evaluations. $\mathcal{L}_{GP_1}$ is the gradient penalty term for applying the Lipschitz Constraint: $\mathcal{L}_{GP_1} = \lambda(||\nabla_{\tilde{x}} D_1||_2 - 1)^2$ where the $\tilde{x}$ is the linear interpolation of the fake feature $\hat{x}$ and the real feature x. The last two D_1 loss functions calculate Wasserstein distance of the fake and the real visual features,

3.2.2 Text Feature Generation Network

Similar to the visual feature generation network, our proposed inverse Text Feature Generation Network $I = \{G_2, D_2\}$ consists of a text feature generator G_2 and a text feature discriminator D_2. The main contribution of this model to the overall architecture is to provide reconstructed text features for calculating cycle-consistency loss, and further regularize the text embedding layer to generate accurate text features by introducing text feature classification loss.

Text Feature Generator G_2: Given the synthesized image features $\hat{x}$ from G_1, the goal of G_2 is to generate realistic text features. The input visual features are around 3500 dimensions, concatenated with a 100 dimension random noise $z \in \mathbb{R}^z$ sampled from Gaussian distribution $N(0, 1)$. The concatenated vectors are then fed into two fully connected layers together with Leaky-ReLU and Tanh

activation functions respectively. The synthesized text features $\hat{\alpha} = G_2(\hat{x}, z, \delta)$ are so far prepared for applying cycle-consistent constraint. The objective function of text feature generator G_2 can be formulated as:

$$\begin{aligned}\mathcal{L}_{G_2} = -\mathbb{E}_{z\sim p_z}[D_2(G_2(\hat{x}, z, \delta), \zeta)] \\ +\mathcal{L}_{cls_2}(G_2(\hat{x}, z, \delta), \zeta),\end{aligned} \tag{3}$$

where the D_2 and the $\mathcal{L}_{cls_2}$ represent Wasserstein loss [3] and the text feature classification loss corresponding to category labels, respectively. δ and ζ represents the weights in G_2 and D_2.

Text Feature Discriminator D_2: Once the reconstructed text features are mapped back from synthesized visual features, they are processed through a fully connected layer with ReLU activator. Afterwards, same as D_1 we simply use a fully connected layer for distinguishing the text feature fidelity and another fully connected layer to classify the text features into different categories. The objective function of D_2 is defined as:

$$\begin{aligned}\mathcal{L}_{D_2} = \frac{1}{2}(\mathcal{L}_{cls_2}(G_2(\hat{x}, z, \delta)) + \mathcal{L}_{cls_2}(s)) + \mathcal{L}_{GP_2} \\ +\mathbb{E}_{z\sim p_z}[D_2(G_2(\hat{x}, z, \delta), \zeta)] - \mathbb{E}_{x\sim p_{data}}[D_2(x, \zeta)],\end{aligned} \tag{4}$$

where the first two terms are text feature classification losses, which enforce the text feature embedding to be as well discriminative as the visual features. $\mathcal{L}_{GP_2}$ is the gradient penalty computed with the same manner of the visual feature discriminator and the last two terms are Wasserstein distance of the fake and the real visual features,

3.2.3 Cycle-Consistency Loss

In theory, learning a forward mapping and a inverse mapping by adversarial losses is able to produce outputs identically distributed as real features [12]. However, even if the forward mapping is conditioned on the seen semantic features, there is no guarantee that the synthesized visual features capture textual features. In order to address this issue, we introduce cycle-consistency loss $\mathcal{L}_{cyc}$ to regularize the visual feature generator G_1 being able to synthesize visual features with semantic information preserved. Once the reconstructed text features are generated by G_2, the cycle consistency loss is computed to update weights on both G_1 and G_2. We also argue that the cycle-consistency loss in our architecture promote the text encoder ψ as well, which is included in G_1. It is defined as:

$$\mathcal{L}_{cyc} = \lambda \frac{1}{N^b} \sum_{n=1}^{N^b} ||G_2(G_1(\alpha, z, \theta), z, \delta) - s||^2, \tag{5}$$

where λ is the coefficient, the N^b denotes the batch size, and s is the text feature from text encoder $\psi(\alpha)$. The cycle-consistency loss is essentially the mean squared error between the reconstructed textual features from G_2 and the real ones directly extracted from noisy text descriptions α.

3.3. Training Procedure

To train the overall model, we consider visual-semantic feature pairs as joint observation. Visual features are either generated by our visual feature generator G_1 or from the ground truth provided in the datasets, whereas text features are either generated by our text feature generator G_2 or encoded by the fully connected layer in G_1. The visual features and text features are further introduced in section 4.1. We train two discriminators D_1 and D_2 separately to distinguish the reality and classify the object category of the synthesized visual features and text features respectively. In each training iteration, two discriminators are updated 5 times, and the both generators are optimized for 1 step. In addition, we follow a training technique from [43] that regularize the generated visual features to be consistent with the cluster centre of the corresponding object class. Lastly, the cycle-consistency loss are applied once in each iteration by calculating the gap between the generated text features and the text features that directly extracted from natural language.

3.4. Zero-Shot Recognition

Given unseen semantic descriptions and random noise z from Gaussian distribution, our trained model can synthesize infinite number of visual features with randomly sampled z. The process can be formulated as following:

$$x_u = G_1(\alpha_u, z, \theta) \tag{6}$$

Once the visual features are synthesized, the zero-shot learning problem becomes a traditional supervised classification problem. For simplicity, we adopt k-nearest neighbor algorithm (K-NN) to conduct this supervised task.

4. Experiments

4.1. Experiment Setting

Datasets: We conduct experiments on two bird datasets: CUB-200-2011 (CUB) [34] and North America Birds (NAB) [33]. The CUB dataset consists of 11,788 images from 200 bird species, and NAB is a significantly larger dataset of birds with 1011 categories and 48,562 images. NAB dataset forms a hierarchy of bird classes, including 555 leaf nodes and 456 parent nodes. The images in NAB are associated with leaf nodes. Elhoseiny *et al.* [11] captioned both datasets with the Wikipedia article. Due to the lack in the Wikipedia articles for some subtle division of classes, some subclasses are merged and 404 classes of birds are yielded with corresponding Wikipedia articles.

Besides, there are two splitting designs according to the relationship between seen categories and unseen categories: Super-Category-Shared splitting (SCS) and Super-Category-Exclusive splitting(SCE). In SCS, unseen categories are chosen to share same super-class with the seen categories. In result, the relevance in this design between seen categories and unseen categories is relatively high. In contrast, all categories in SCE belong to the same super-class are split into either seen or unseen categories. Intuitively, zero-shot recognition performance should be better in SCS-split than SCE-split. Conventional ZSL methods[1, 2, 28, 30] use SCS-split only, whereas we use both splits to validate our approach.

Text Feature: Elhoseiny *et al*. [11] collected raw Wikipedia articles for CUB and NAB datasets. They first tokenized the Wikipedia articles into words and got rid of the full stops. In order to weight the terms in the dataset appropriately, Term Frequency-Inverse Document Frequency (TF-IDF) [32] is adopted to extract text feature vectors. The dimensionality of TF-IDF feature for dataset CUB [34] and NAB [33] are 11,083 and 13,585 respectively. However, the TF-IDF is further embedded into a lower dimension textual representations for suppressing the noise.

Visual Feature: Elhoseiny *et al*. [11] prepared visual features from images in the two bird datasets with Visual Parts CNN Detector/Encoder (VPDE). Input images are reshaped to 224×224 and detected by fast-RCNN framework with VGG16. The detected parts are then fed to the VPDE network, where 512-dimensional feature vectors are extracted for each semantic part. For dataset CUB, seven semantic parts are used to train the VPDE network. Due to the lack of annotations for the "leg" part in the NAB dataset, we use only six visual parts without the "leg" part. The dimensionality of visual features for CUB and NAB are 3582 and 3072 respectively.

Implementation Details: Theoretically, the synthesized text features from the inverse GAN provides multi-modal constraint information to optimize the forward visual feature generation model. Thus, if the reconstructed text features from inverse GAN are accurate throughout the training process of the forward GAN, the cycle-consistent training should not only converge stably but also faster.

However, there are three reasons why we choose not to pretrain our inverse model. First, note that our proposed cycle architecture has a low complexity with only several fully connected layers. Even if we pretrain the inverse model, the convergence comes nearly same as training the whole model from scratch. Second, with the classification information introduced in both discriminators, the model usually finds the optimal gradient extremely quickly. Last but not least, with a large λ for cycle-consistency loss (we set as 10 in our experiments), the forward and the inverse networks both promote each other collaboratively.

Table 1. Top-1 accuracy (%) on CUB and NAB datasets with two split settings.

	CUB		NAB	
Methods	SCS	SCE	SCS	SCE
MCZSL[1]	34.7	-	-	-
WAC-Linear[10]	27.0	5.0	-	-
WAC-Kernel [9]	33.5	7.7	11.4	6.0
ESZSL [30]	28.5	7.4	24.3	6.8
SJE [2]	29.9	-	-	-
ZSLNS [28]	29.1	7.3	24.5	6.8
SynC$_{fast}$ [4]	28.0	8.6	18.4	3.8
SynC$_{OVO}$ [4]	12.5	5.9	-	-
ZSLPP[11]	37.2	9.7	30.3	8.1
GAZSL[43]	43.7	10.3	35.6	8.6
CANZSL	**45.8**	**14.3**	**38.1**	**8.9**

In order to compare our approach with GAZSL [43] to show the superiority of our cycle-consistent architecture, we follow the same setting in the forward visual feature generation GAN as GAZSL. The batch size is fixed as 1000, and the learning rate, as shown in Algorithm 1, is set as 0.0001. We use Adam optimizer [19] with β_1 as 0.5 and β_2 as 0.9 respectively. All experiments are conducted on a server with 16 Intel(R) Xeon(R) Gold 5122 CPUs and 2 GeForce RTX 2080 Ti GPUs.

4.2. Performance evaluation

Experiments are conducted on both SCE and SCS splits of the two bird datasets CUB and NAB to evaluate our approach. We compare our approach with other eight state-of-the-art algorithms: GAZSL [43], ZSLPP[11], SynC[4], ZSLNS[28], SJE[2], ESZSL[30], WAC[9], MCZSL[1]. The source code of GAZSL, ZSLPP, ESZSL, and ZSLNS are available online. For the rest of methods, we directly cite the highest scores reported in [43]. For the attribute-based methods, we simply replace the attributes input with the textual features. ZSLPP and MCZSL extracts visual features from the semantic parts of birds. MCZSL simply adopted annotated semantic parts to supervise visual feature extraction during the testing stage. In comparison, our approach, ZSLPP and GAZSL used detected semantic parts in both training and testing phase. As a result, the performance of the final zero-shot classification is expected to degrade due to less accurate detection of semantic parts compared to manual annotation in MCZSL. Table 1 demonstrates the performance comparisons on CUB and NAB datasets. Generally, our method consistently outperforms the state-of-the-art methods. On the conventional split setting (SCS), our approach outperforms the runner-up (GAZSL) by a considerable gap: 2.1% and 2.5% on CUB dataset and NAB dataset, respectively. However, ZSL on

Table 2. Effects of different components on zero-shot classification accuracy (%) on CUB and NAB datasets with SCS split setting.

	CUB		NAB	
Methods	Text feature	TF-IDF	Text feature	TF-IDF
CYC-only	45.1	44.8	36.5	35.9
ADV-CYC-only	45.5	45.2	37.2	37.1
CLA-CYC-only	45.3	44.9	37.1	36.7
CANZSL	**45.8**	**45.5**	**38.1**	**37.3**

SCE-split remains rather challenging. The fact that there is less relevant information between the training and testing set makes it extremely hard to transfer knowledge from seen classes to unseen classes. Although our method improves the performance by 4% on the CUB dataset, the improvement on NAB is merely 0.3%. We will show a higher improvement on the general merit of ZSL in Sec 4.5 with two split settings.

4.3. Ablation Study

We now report the ablation study of the effect of the cycle-consistency loss, the classification loss and the adversarial loss in the inverse GAN. We trained three variants of our model by only keeping the cycle-consistency loss, adversarial loss and classification loss, denoted as CYC-only, ADV-CYC-only, and CLA-CYC-only, respectively. In the case of CYC-only, the textual feature generator is merely updated by the cycle-consistency loss, whereas ADV-CYC is optimized by the cycle-consistency loss as well as an adversarial loss from the discriminator. Similarly, the CLA-CYC-only variant is optimized by the cycle-consistency loss and the classification loss.

Table 2 shows the performance of each setting. It is clear that each component significantly contributes to the overall architecture. We also observe that with any proposed component, the performance of each variants is much higher than the runner-up method GAZSL shown in Table 1, which demonstrates the importance of each component. We argue that the adversarial loss and the classification loss are critically complementary to each other. The cycle-consistency loss can only ensure the mapping from synthesized textual features are accurately corresponding to the extracted ones from visual feature generation network. However, with the adversarial loss and classification loss applied on the pseudo textual feature generator, the text encoder in visual feature synthesis network is beneficial from the cycle-consistency loss by being forced to adapt to class label information.

We investigate whether the cycle-consistency loss should be applied on the textual feature noisy or the TF-IDF representation of text description. As shown in Table 3, generally our method with cycle-consistency loss applied on textual features outperforms the one with cycle-consistency loss applied on noisy text TF-IDF representations.

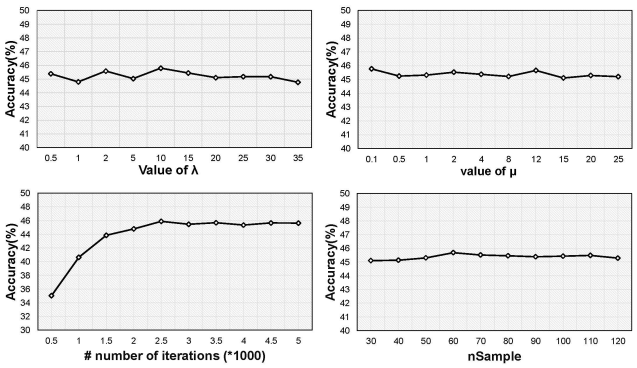

Figure 3. Parameters sensitivity of the proposed method.

4.4. Parameters Sensitivity

In order to investigate the most appropriate hyper-parameter values for the proposed CANZSL model, we compare and demonstrate the performance with various hyper-parameter values. In our cycle architecture, we argue that the cycle-consistency loss is the most significant component according to the performance demonstrated in the ablation study. Even if the forward generator and the inverse generator are merely updated by cycle-consistency loss, we can outperform our baseline GAZSL [43] by 1.4%. Here we demonstrate the performance comparison between various coefficients λ for cycle-consistency loss. It is shown at the upper-left on Fig. 3 that when the λ increases from 0.5 to 10, the performance is extremely unstable, and it decreases slowly when λ is greater than 10. Intuitively, we argue that 10 is the best coefficient for cycle-consistency loss.

We also experimented on several different values μ for the coefficient of classification loss in the inverse network. Interestingly, there is no obvious trend as λ in the upper-right graph on Fig. 3. As a result, our model is not sensitive to the hyper-parameter of μ. Intuitively, we adopt the value 12 with the highest performance.

Even if our CANZSL model only involves a number of fully connected layers, which guarantee that the training usually converges drastically. From the lower-left line chart in Fig. 3, we can see the performance reaches 45.8% merely in 2500 iterations.. Afterwards, the performance keeps stable with the iteration goes up to 5000.

In testing phase, we uses different sampling numbers to evaluate our trained model. The result is shown in the lower-right in Fig. 3. We yield best performance when sampling 60 visual features.

4.5. Results of the Generalized ZSL

The conventional zero-shot learning categorizes text samples into unseen classes without seen classes in test

Table 3. The performances (in %) of the generalized ZSL on CUB and NAB datasets with two split settings.

	CUB		NAB	
Methods	SCS	SCE	SCS	SCE
WAC-Linear[10]	23.9	4.9	23.5	-
WAC-Kernel [9]	22.5	5.4	0.7	2.3
ESZSL [30]	18.5	9.2	24.3	2.9
ZSLNS [28]	14.7	4.4	9.2	2.3
SynC$_{fast}$ [4]	13.1	4.0	2.7	0.8
SynC$_{OVO}$ [4]	1.7	1.0	0.1	-
ZSLPP[11]	30.4	6.1	12.6	3.5
GAZSL[43]	35.4	8.7	20.4	5.8
CANZSL	**40.2**	**12.5**	**25.6**	**6.8**

phase, whereas the seen classes are usually more common than unseen classes. In this case, it is unrealistic to assume we will never encounter unseen objects during the test phase [5]. Chao *et al*. [5] recently proposed a more appropriate metric called Area Under Seen-Unseen accuracy Curve (AUSUC) that can evaluate generalized zero-shot learning (GZSL) approaches, by acknowledging that there is an inherent trade-off between recognizing seen classes and recognizing unseen classes.

In order to compare with the runner-up approach GAZSL, we directly cite the performance results reported in [43]. We show the AUSUC results on both SCS split and SCE split in Table 3 and we observe that the proposed CANZSL approach performs particularly competitive under the more realistic generalized ZSL task. On dataset CUB and NAB and corresponding splits, our CANZSL obtains superior performance with a large margin against the competitors. The result indicates that our approach performs much better than other competitors on alleviating the issue of the seen-unseen bias under the generalized ZSL scenario. In other words, the proposed approach can improve the performances of unseen classes while maintaining the performances of seen classes.

4.6. t-SNE Demonstration

Fig. 4 demonstrates the t-SNE [22] visualization of the real visual features and the synthesized visual features from unseen classes on CUB dataset. From the real samples in Fig. 4(a) we can see that some categories overlap with each other by a large degree, such as *black billed cuckoo* and *yellow billed cuckoo*. The overlapping also exists in the synthesized features as shown in Fig. 4(b). It is reasonable when considering these two bird classes only differentiate in colour. This insight observation indicates that the underlying data distribution is well captured in our model. Also, thanks to class label information involved, the synthesized features are extremely discriminative as they obviously distribute in separate clusters.

(a) Real visual features (b) Synthesized visual features

Figure 4. t-SNE visualization of features from random 10 unseen classes on CUB dataset. Each color represents a specific class label indicated on the right hand side of (b). (a) visualizes the visual representations directly extracted from the visual feature extractor E. (b) shows the visual features that inferenced from the trained forward GAN.

4.7. Results of ZSL from Attributes

As discussed in Sec. 2.1, it is more practical to conduct zero-shot learning from natural language. Further, we reported our superior performance on this task. In theory, during embedding the text description into textual features, it is unlikely to preserve non-trivial information thoroughly. Hence, in the same setting, our method should be able to perform better in zero-shot learning from attributes, which perfectly preserve all useful information. For zero-shot learning from attributes, the fully connected layer, which was adopted to suppress the text noise, is removed from the visual feature generator.

We demonstrate our performance in ZSL from attributes, and compare with four other state-of-the-art methods in Table 4. From the Table, we can notice that the proposed method outperforms the state-of-the-art methods not only in ZSL from noisy text but from attributes as well.

Table 4. The performances (in %) of ZSL from attribute-based semantic representation on CUB dataset.

Input	Noisy Text	Attributes
ESZSL [30]	28.5	53.9
SJE [2]	29.9	53.9
SynC [4]	28.0	55.6
GAZSL[43]	43.7	55.8
CANZSL	**45.8**	**56.5**

5. Conclusions

In this paper, we proposed novel cycle-consistent adversarial networks for ZSL from natural language, which leverage multi-modal cycle-consistency loss to regularize the visual feature generator to preserve semantics during training. An inverse GAN is added to reconstruct visual features back to textual representations. Experiments showed that our approach consistently performs favorably against the

state-of-the-art methods not only on traditional ZSL, but on generative ZSL as well, with an outstanding capability of visual feature generation. We also showed in an ablation study that the adversarial loss, classification loss and cycle-consistency loss can promote the overall architecture collaboratively. Furthermore, we validated that our CANZSL is also able to perform well on the task of the ZSL from attributes. In our future work, we will also study how to optimize the testing phase and utilize the unseen class descriptions during training procedure.

References

[1] Z. Akata, M. Malinowski, M. Fritz, and B. Schiele. Multi-cue zero-shot learning with strong supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 59–68, 2016.

[2] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2927–2936, 2015.

[3] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[4] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha. Synthesized classifiers for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5327–5336, 2016.

[5] W.-L. Chao, S. Changpinyo, B. Gong, and F. Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *European Conference on Computer Vision*, pages 52–68. Springer, 2016.

[6] L. Chen, H. Zhang, J. Xiao, W. Liu, and S.-F. Chang. Zero-shot visual recognition using semantics-preserving adversarial embedding networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1043–1052, 2018.

[7] T.-H. Chen, Y.-H. Liao, C.-Y. Chuang, W.-T. Hsu, J. Fu, and M. Sun. Show, adapt and tell: Adversarial training of cross-domain image captioner. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 521–530, 2017.

[8] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494, 2015.

[9] M. Elhoseiny, A. Elgammal, and B. Saleh. Write a classifier: Predicting visual classifiers from unstructured text. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2539–2553, 2017.

[10] M. Elhoseiny, B. Saleh, and A. Elgammal. Write a classifier: Zero-shot learning using purely textual descriptions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2584–2591, 2013.

[11] M. Elhoseiny, Y. Zhu, H. Zhang, and A. Elgammal. Link the head to the beak: Zero shot learning from noisy text description at part precision. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

[12] 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*, pages 2672–2680, 2014.

[13] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[14] Y. Guo, G. Ding, J. Han, and Y. Gao. Synthesizing samples fro zero-shot learning. IJCAI, 2017.

[15] Y. Guo, G. Ding, J. Han, and Y. Gao. Zero-shot learning with transferred samples. *IEEE Transactions on Image Processing*, 26(7):3277–3290, 2017.

[16] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[17] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017.

[18] H. Jiang, R. Wang, S. Shan, Y. Yang, and X. Chen. Learning discriminative latent attributes for zero-shot classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4223–4232, 2017.

[19] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[20] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958. IEEE, 2009.

[21] J. Lei Ba, K. Swersky, S. Fidler, et al. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4247–4255, 2015.

[22] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[23] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[24] X. Mao, Q. Li, H. Xie, R. Y. Lau, and Z. Wang. Multi-class generative adversarial networks with the l2 loss function. *arXiv preprint arXiv:1611.04076*, 5, 2016.

[25] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[26] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*, 2013.

[27] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2642–2651. JMLR. org, 2017.

[28] R. Qiao, L. Liu, C. Shen, and A. Van Den Hengel. Less is more: zero-shot learning from online textual documents with noise suppression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2249–2257, 2016.

[29] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[30] B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. In *International Conference on Machine Learning*, pages 2152–2161, 2015.

[31] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[32] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[33] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, and S. Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. pages 595–604, 06 2015.

[34] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[35] L. Wu, Y. Wang, and L. Shao. Cycle-consistent deep generative hashing for cross-modal retrieval. *IEEE Transactions on Image Processing*, 28(4):1602–1612, 2019.

[36] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5542–5551, 2018.

[37] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1316–1324, 2018.

[38] A. Zeynep, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for image classification. In *IEEE transactions on pattern analysis and machine intelligence*, volume 38, pages 1425–1438. IEEE, 2016.

[39] J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.

[40] Z. Zhong, L. Zheng, Z. Zheng, S. Li, and Y. Yang. Camera style adaptation for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5157–5166, 2018.

[41] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision*, pages 597–613. Springer, 2016.

[42] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017.

[43] Y. Zhu, M. Elhoseiny, B. Liu, X. Peng, and A. Elgammal. A generative adversarial approach for zero-shot learning from noisy texts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1004–1013, 2018.