A Novel Baseline for Zero-shot Learning via Adversarial Visual-Semantic Embedding

Yu Liu
yu.liu@esat.kuleuven.be
Tinne Tuytelaars
tinne.tuytelaars@esat.kuleuven.be

ESAT-PSI
KU Leuven
Leuven, Belgium

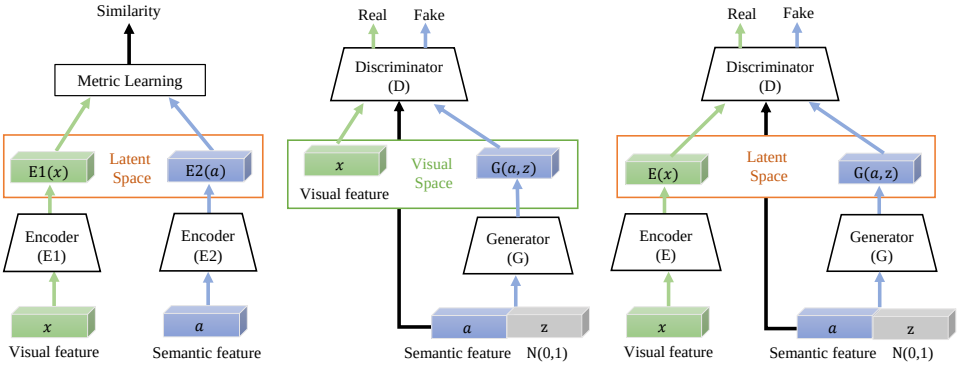
Abstract

Zero-shot learning (ZSL) has been attracting ever-increasing research interest due to its capability of recognizing novel or unseen classes. A lot of studies on ZSL are based mainly on two baseline models: compatible visual-semantic embedding (CVSE) and adversarial visual feature generation (AVFG). In this work, we integrate the merits of the two baselines and propose a novel and effective baseline model, coined *adversarial visual-semantic embedding* (AVSE). Different from CVSE and AVFG, AVSE learns visual and semantic embeddings adversarially and jointly in a latent feature space. Additionally, AVSE integrates a classifier to make latent embeddings discriminative, and a regressor to preserve semantic consistency during the embedding procedure. Moreover, we perform embedding-to-image generation which visually exhibits the embeddings learned in AVSE. The experiments on four standard benchmarks show the advantage of AVSE over CVSE and AVFG, and empirical insights through quantitative and qualitative results. Our code is at https://github.com/Liuy8/AVSE.

1 Introduction

Nowadays, many computer vision tasks [7, 20, 21, 28] achieve outstanding performance under the standard supervised learning scenario, where the pre-defined classes are *consistent* at both training and test time. In contrary, in the setting of *zero-shot learning* (ZSL) [17, 26], the test classes are *disjoint* with the training classes. In other words, ZSL targets at classifying *novel/unseen* classes outside training classes. Apart from the above standard ZSL setting, some research is also focused on an extended yet practical setting, namely *generalized zero-shot learning* (GZSL) [37], where both seen and unseen classes are included in the test set. The problem of ZSL or GZSL has been much researched in the literature for a wide variety of real-world applications such as person re-identification [36], event detection [5] and visual navigation [3].

While no visual data is available for those unseen classes, they are also pre-defined in a form of auxiliary *semantic features*, for example, common attributes [17], word embedding [24], sentence descriptions [29] or semantic similes [22]. The key to solving ZSL is firstly learning a model to bridge visual and semantic features from seen classes only, and then transferring the model to handle unseen classes at test time. In general, most ZSL approaches build on top of one out of two baseline models. (1) The first baseline (Fig. 1 (a))



(a) CVSE model (b) AVFG model (c) AVSE model (ours)

Figure 1: Conceptual illustration of three ZSL baseline models. (a) CVSE: measuring the compatibility between visual and semantic embedding in the *latent space*; (b) AVFG: using semantic features to generate visual features in the *visual space*; (c) our AVSE: performing adversarial learning between visual and semantic embedding in the *latent space*.

is *compatible visual-semantic embedding* (CVSE) [1, 34, 38, 43], which projects both visual and semantic features into a latent embedding space and then learns a metric learning objective to maximize the compatibility between the two embeddings. The latent embeddings in CVSE are able to retain high discrimination and bridge the visual-semantic gap. (2) The second baseline (Fig. 1 (b)), namely *adversarial visual feature generation* (AVFG) [32, 39, 44], is inspired by generative adversarial network (GAN). The generator takes as input semantic features and adversarially generates pseudo visual features as similar as possible to real ones. It addresses the lack of visual samples for unseen classes, and thereby converts ZSL to a conventional supervised learning problem.

In this work, we present a novel ZSL baseline (Fig. 1 (c)), coined *adversarial visual-semantic embedding* (AVSE), which allows to integrate the merits from CVSE and AVFG. First, like CVSE, AVSE also maps visual and semantic features into a latent embedding space to overcome the modality gap. However, the embedding in AVSE are learned in an adversarial fashion, instead of utilizing metric learning functions. Second, in contrast to AVFG which learns to generate the input visual features, AVSE introduces a visual encoder to embed the visual features in a latent embedding space. We find that the visual encoder helps to balance the capabilities between the generator and discriminator, and to ease the following feature generation from semantic features. As shown in the full model of our AVSE (Fig. 2), we further learn a *classifier* on top of the visual-semantic embedding to maintain their discriminative power. Additionally, a *regressor* is imposed to preserve the semantic consistency with the original semantic features. The entire AVSE model is end-to-end trainable by jointly optimizing several objective functions. Moreover, to provide new insights into our embedding, we train an *embedding-to-image* generator and visually demonstrate what visual information is preserved or discarded by the model (Sec. 3.4). Lastly, we experiment with several standard benchmark datasets where extensive results demonstrate the improvements of AVSE beyond CVSE and AVFG, under the context of both ZSL and GZSL. Additional quantitative and qualitative results shed further light on the effectiveness of the components used in AVSE.

2 Background: ZSL Baseline Models

The problem of ZSL has become a significant and realistic scenario in many vision applications [3, 5, 36, 37]. In addition, some variants have developed over the past ten years, including generalized ZSL, transductive ZSL and generative ZSL. The next summarizes the baseline models upon which previous approaches are mainly built.

CVSE Baseline Model. Figure 1 (a) depicts a general pipeline of CVSE. Metric learning objectives, such as hinge loss or contrastive loss, are optimized to preserve the compatibility between paired visual-semantic embeddings in the latent space. Latent embeddings are able to distill the most common and important information across two modalities, which help to make the model generalize better to unseen classes. In recent years, many works have improved CVSE with deep neural networks and solved the task in an end-to-end fashion [1, 4, 6, 13, 16, 23, 34, 35, 38, 42, 43]. For instance, the work of [34] presented a bidirectional latent embedding learning framework with two subsequent learning stages including bottom-up learning and top-down learning. In [41], it extensively studied the combination of several compatibility functions. LATEM [38] constructed several latent embedding spaces and combined multiple compatibility scores. In other works [13, 35], the concept of knowledge graph is further adopted to leverage structure information among latent embeddings.

AVFG Baseline Model. Deep generative models [10, 15, 19] have recently attracted much attention in the research of ZSL due to their ability of synthesizing new visual features for unseen classes [8, 18, 32, 39, 44]. This allows to address ZSL under a supervised learning mode. Figure 1 (b) shows the pipeline of AVFG, where the generator is trained to produce fake visual features $G(\mathbf{a},\mathbf{z})$ conditioned on semantic feature $\mathbf{a}$ and random noise $\mathbf{z}$. Meanwhile, the discriminator is learned to retain the ability of telling real and fake visual features apart. For example, f-CLSWGAN [39] built a GAN model [10] to produce CNN visual features from random noise conditioned by semantic features. Similarly, Cycle-CLSWGAN [8] added a cycle-consistency loss to preserve semantic consistency in synthetic visual features. To ensure that fake samples were close to real ones, the recent work Lis-GAN [18] defined soul samples to regularize the generator.

Comparison. As shown Figure 1 (c), our AVSE combines the latent embedding in CVSE and the feature generation in AVFG. In contrast to AVFG, we perform the adversarial learning in the latent space, by using a visual encoder to learn visual embedding $E(\mathbf{x})$, and making the synthesized semantic embedding $G(\mathbf{a},\mathbf{z})$ as similar as possible to the visual embedding. The components in AVSE are pre-existing, however, being able to leverage those components to design a generic baseline model is precisely important for the field of ZSL.

3 Adversarial Visual-Semantic embedding

Notation and Definition. In the setting of ZSL, the full set of classes $\mathcal{Y}$ is divided into *seen* classes $\mathcal{Y}^s$ and *unseen* classes $\mathcal{Y}^u$, so that $\mathcal{Y}^s \cup \mathcal{Y}^u = \mathcal{Y}$ and $\mathcal{Y}^s \cap \mathcal{Y}^u = \emptyset$. Regarding the seen classes, there is a training set $\mathcal{D}^s = \{(\mathbf{x}_i, y_i, \mathbf{a}_i), i = 1,\ldots,N^s\}$, where $\mathbf{x}_i \in \mathcal{X}^s$ is the visual representation extracted from a pre-trained CNN and $\mathbf{a}_i \in \mathcal{A}^s$ indicates the semantic feature (*e.g.* an attribute vector) for the seen class $y_i \in \mathcal{Y}^s$. Similarly, the data set for the unseen classes is denoted as $\mathcal{D}^u = \{(\mathbf{x}_i, y_i, \mathbf{a}_i), i = 1,\ldots,N^u\}$, where $\mathbf{x}_i \in \mathcal{X}^u$, $y_i \in \mathcal{Y}^u$ and $\mathbf{a}_i \in \mathcal{A}^u$. Note that the visual features $\mathcal{X}^u$ are not available during training. In the next, we drop the subscript i for simplicity. The classical ZSL aims to learn a function $f : \mathbf{x} \rightarrow \mathcal{Y}^u$, while GZSL classifies both seen and unseen samples by $f : \mathbf{x} \rightarrow \mathcal{Y}^s \cup \mathcal{Y}^u$.

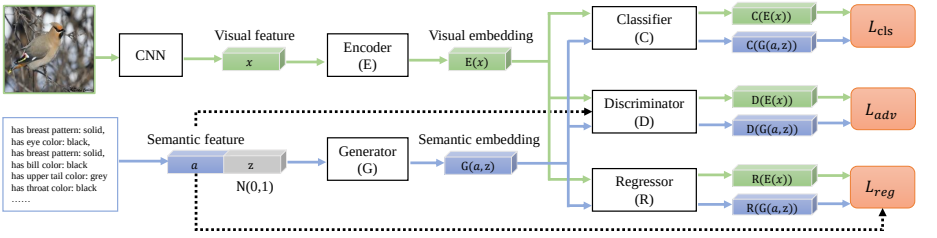


Figure 2: Network architecture of the proposed adversarial visual-semantic embedding (AVSE). The adversarial objective is performed between visual and semantic embeddings within a latent space. The classifier makes the embedding discriminative, and the regressor is to preserve their semantic consistency with the semantic features.

3.1 Model Architecture

The overall framework of AVSE is illustrated in Fig. 2, which is optimized jointly by three objectives: an adversarial objective, a classification objective and a regression objective. First of all, the adversarial objective is to make the synthesized semantic embedding consistent with the distribution of the visual embedding. In addition, we train a classifier to supervise the discriminative power of visual and semantic embeddings. Finally, the regressor reconstructs the input semantic feature so as to preserve semantic consistency while learning the embeddings. Next, we introduce details on these three objectives.

Adversarial objective. We extract the input visual feature $\mathbf{x}$ from a pre-trained CNN and then use the visual encoder E to learn the visual embedding $E(\mathbf{x})$. The generator G learns to synthesize the semantic embedding $G(\mathbf{a},\mathbf{z})$ conditioned on the concatenation of the semantic feature $\mathbf{a}$ and random noise $\mathbf{z} \sim \mathcal{N}(0,1)$. Random noise helps to retain the diversity of synthetic samples, as each semantic feature can describe a variety of image instances. Then, we concatenate either $E(\mathbf{x})$ or $G(\mathbf{a},\mathbf{z})$ with $\mathbf{a}$ before passing them to the discriminator D. Specifically, we consider the visual embedding $E(\mathbf{x})$ as real samples and the semantic embedding $G(\mathbf{a},\mathbf{z})$ as the fake samples. Like f-CLSWGAN [39], we employ the WGAN-GP [11] to train the generator and the discriminator jointly. The adversarial objective for learning visual and semantic embeddings is formulated with

$$\mathcal{L}_{adv}(E,G,D) = \mathbb{E}[D(E(\mathbf{x}),\mathbf{a})] - \mathbb{E}[D(G(\mathbf{a},\mathbf{z}),\mathbf{a})] - \lambda\mathbb{E}[(||\nabla_{\mathbf{e}}D(\mathbf{e},\mathbf{a})||_2 - 1)^2], \quad (1)$$

where $\mathbf{e} = \mu E(\mathbf{x}) + (1-\mu)G(\mathbf{a},\mathbf{z})$ with $\mu \sim U(0,1)$ and the penalty coefficient $\lambda = 10$.

Classification objective. The above adversarial objective enables to bridge the distributions between visual and semantic embeddings. Nevertheless, we still need to supervise the discrimination of those embeddings. To this end, we learn a shared classifier C on top of both $E(\mathbf{x})$ and $G(\mathbf{a},\mathbf{z})$. We train the classification objective with the negative log likelihood by

$$\mathcal{L}_{cls}(E,G,C) = -\mathbb{E}[\log P(y|E(\mathbf{x}))] - \mathbb{E}[\log P(y|G(\mathbf{a},\mathbf{z}))], \quad (2)$$

where y is the ground-truth class label. The Softmax probabilities $P(y|E(\mathbf{x}))$ and $P(y|G(\mathbf{a},\mathbf{z}))$ are compute by

$$P(y|E(\mathbf{x})) = \frac{\exp\left(C(E(\mathbf{x}))_y\right)}{\sum_{y_k \in \mathcal{Y}^s} \exp\left(C(E(\mathbf{x}))_{y_k}\right)}, P(y|G(\mathbf{a},\mathbf{z})) = \frac{\exp\left(C(G(\mathbf{a},\mathbf{z}))_y\right)}{\sum_{y_k \in \mathcal{Y}^s} \exp\left(C(G(\mathbf{a},\mathbf{z}))_{y_k}\right)}, \quad (3)$$

Notice that the classification objective is performed for seen classes only, as we do not have the labelled image data of unseen classes.

Regression objective. Furthermore, we add a regressor to map the visual and semantic embeddings back to the semantic feature space. The reconstructed semantic features, $R(E(\mathbf{x}))$ and $R(G(\mathbf{a},\mathbf{z}))$, help preserving the semantic consistency of our embedding. To train the regressor, we adopt the L1 norm loss

$$\mathcal{L}_{reg}(E,G,R) = \mathbb{E}[||R(E(\mathbf{x})) - \mathbf{a}||_1] + \beta\mathbb{E}[||R(G(\mathbf{a},\mathbf{z})) - \mathbf{a}||_1], \tag{4}$$

where β is the weight for the second term.

Full objective. Our full objective integrates the above three objectives simultaneously

$$\mathcal{L}(E,G,D,C,R) = \mathcal{L}_{adv}(E,G,D) + \gamma\mathcal{L}_{cls}(E,G,C) + \mathcal{L}_{reg}(E,G,R), \tag{5}$$

where γ regulates the relative importance of the classifier term. Finally, the goal is to optimize all the parameters in the model based on

$$E^*,G^* = \arg\min_{E,G,C,R}\max_{D}\mathcal{L}(E,G,D,C,R). \tag{6}$$

3.2 Discussion

Despite that AVSE is a simple and generic baseline by integrating pre-existing components, it has technical strength over previous approaches. In particular, generative ZSL methods [18, 39] have an inherent imbalance between the capabilities of generator and discriminator. Specifically, the discriminator has been able to tell real and fake samples, while the generator is still struggling in pushing fake samples consistent with real samples distribution that is fixed by the choice of a given dataset. To solve the imbalance issue, AVSE leverages a simple yet effective visual encoder (E) which can progressively adjust the real samples to make the capability of the generator fit properly with that of the discriminator. Consequently, balancing the capabilities helps to improve the generation quality. For instance, AVSE outperforms other methods significantly even when only one sample is generated (Fig. 5). One related work to ours is Tempered Adversarial Networks [30] that add a lens in between the real data and the discriminator. However, their work is focused on general image generation, rather than ZSL image classification.

3.3 Learning Classifiers for Unseen Classes

In the test stage, we use the trained AVSE model to generate semantic embeddings $G(\mathbf{a},\mathbf{z})$ for unseen classes, denoted as $\mathcal{U} = \{(G(\mathbf{a},\mathbf{z}),y)\}$, where $\mathbf{a} \in \mathcal{A}^u$ and $y \in \mathcal{Y}^u$. Based on the set $\mathcal{U}$, it enables to train classifiers for those unseen classes. (1) For the classical ZSL, we train a Softmax classifier $P(y|\mathbf{h},\theta)$ with $(\mathbf{h},y) \in \mathcal{U}$; (2) For GZSL, we first extract the visual embedding of training images, denoted as $\mathcal{S} = \{(E(\mathbf{x}),y)\}$, where $y \in \mathcal{Y}^s$. Then we merge $\mathcal{U}$ and $\mathcal{S}$ to train a GZSL classifier $P(y|\mathbf{h},\theta)$ with $(\mathbf{h},y) \in \mathcal{U}\cup\mathcal{S}$. In summary, given any test image, we extract and classify its visual embedding $E(\mathbf{x})$ by

$$y^* = \arg\max_{y\in\hat{\mathcal{Y}}} P(y|E(\mathbf{x}),\theta), \tag{7}$$

where $\hat{\mathcal{Y}} = \mathcal{Y}^u$ for ZSL and $\hat{\mathcal{Y}} = \mathcal{Y}^s \cup \mathcal{Y}^u$ for GZSL.

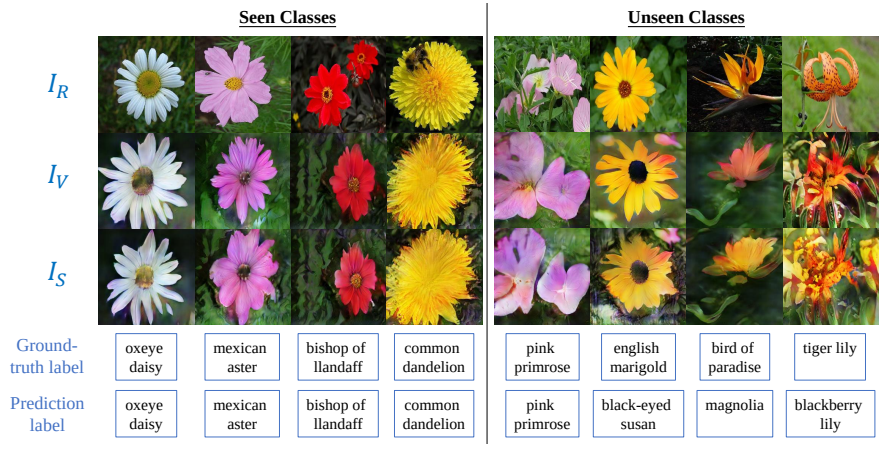


Figure 3: Visualization of our generated images for several seen classes (Left) and unseen classes (Right) from Oxford Flowers [25]. Compared to the real images I_R, the generated images I_V and I_S, which are conditioned on visual and semantic embeddings, respectively, preserve critical semantics about the flowers. Below the images, we show the ground-truth class labels and the predicted ones, including both success and failure cases.

3.4 Visualization of Embeddings

Embedding-to-image generation. To visually show what embeddings are learned in AVSE, we train an image generation network that synthesizes new images conditioned on the embedding. We adapt the StackGAN architecture [40] to perform the *embedding-to-image* generation process. Specifically, we train a new StackGAN based on the visual embedding $E(\mathbf{x})$. Following the ZSL setting, only seen data is used to train the StackGAN, which generates 256×256 photo-realistic images, denoted as I_V. Likewise, we feed the semantic embedding $G(\mathbf{a}, \mathbf{z})$ into StackGAN and study whether their generated images I_S are visually consistent with I_V. Figure 3 visualizes the real images I_R and their corresponding generated images I_V and I_S for Oxford Flowers [25]. It can be seen that I_V and I_S have similar visual content and appearance. We further discuss the details from two aspects:

What information is preserved. By comparing the generated images with the original real images, we can observe what information has been preserved in visual-semantic embeddings. First, the color and shape clues about the flowers are kept in the generated images for both seen and unseen classes. Besides, some attribute parts like the petal and stamen are preserved well, because they are critical for representing the semantics. The generated flowers in the last column are less accurate due to their complex appearance.

What information is discarded. On the other hand, we aim to qualitatively show, based on these visualizations, what information has been discarded by the encoder E and the generator G. Intuitively, these information should not affect the correct recognition of the images. For example, the background in the generated images is blurred compared to the real images. Considering other irrelevant information, the bee in the fourth real image is missing in the generated images; when several flowers appear in the real image, fewer flowers are synthesized in the generated images (in the 3rd and 5th columns).

Dataset	#images (train+val)	#images (test seen/unseen)	#seen classes	#unseen classes	Ratio (seen/unseen)	#Attributes
CUB	7057	1764/2679	150	50	3.00	312
SUN	10320	2580/1440	645	72	8.96	102
AWA	23527	5882/7913	40	10	4.00	85
FLO	1640	5394/1155	82	20	4.10	1024

Table 1: Statistics of the four datasets for zero-shot learning. Additionally, we show the ratio of seen classes to unseen classes in each dataset.

4 Experiments

Datasets and settings. We experiment with four widely-used benchmarks including Caltech-UCSD-Birds (CUB) [33], Oxford Flowers (FLO) [25], SUN Attributes (SUN) [27] and Animals with Attributes2 (AWA) [37]. To ensure the disjoint between seen and unseen classes, we follow the new data splits in [37]. The dataset settings are summarized in Table 4.

Implementation details. We extract visual features from the ResNet-101 [12] model pre-trained on ImageNet. For semantic features, we use the attribute vectors pre-defined in the datasets, except that FLO uses a 1024-dimensional RNN feature from [29]. Besides, the main components in AVSE are implemented with multi-layer perceptron (MLP). To be specific, the encoder E and the generator G contain two fully-connected (FC) layers. Leaky ReLU activation is added behind the first FC layer and standard ReLU activation is used for the second FC layer. The discriminator D is built with an FC layer, a Leaky ReLU layer and an FC layer. In the classifier C, one FC layer is trained based on the number of training classes. Like [39], the classifier is pre-trained before training the entire model. The regressor R contains one FC layer with ReLU activation. The output of R has the same size as the semantic feature. We optimized the AVSE model via Adam [14] with an initial learning rate of 0.001 and a mini-batch size of 64. We set the parameters $\beta = 0.1$ and $\gamma = 0.01$.

Evaluation metrics. Following the metrics used in [37], when evaluating the classical ZSL, we denote the accuracy **T** for unseen classes $\mathcal{Y}_u$; for the GZSL evaluation, we measure the accuracy **S** on seen classes $\mathcal{Y}_s$, the accuracy **U** on unseen classes $\mathcal{Y}_u$, and their harmonic mean **H**, *i.e.* $\mathrm{H} = 2 \times (\mathrm{S} \times \mathrm{U})/(\mathrm{S} + \mathrm{U})$.

Compared methods. First of all, we implement the two baseline models including CVSE and AVFG, and compare AVSE with them. For a fair comparison, the classification and regression objectives are imposed for all the three baselines. In addition, we show other ZSL methods that are widely compared in the literature (Table 2 and Table 3).

4.1 Comparison and Discussion

ZSL results. We report the ZSL results on the four datasets in Table 2. First, AVSE outperforms both CVSE and AVFG consistently, which verifies the effectiveness of integrating latent embeddings and adversarial learning. In addition, generative methods outperform embedding methods, because of generating new and extensive samples for unseen classes. We can see that AVSE is competitive with other generative methods across most metrics. Moreover, AVSE yields the best average accuracy (66.0%) over the four datasets.

GZSL results. Table 3 compares the results under the GZSL setting. Note that, generative methods alleviate the performance gap between the S and U accuracy, while embedding methods have a relatively large gap between the two accuracy results. Likewise, AVSE per-

Method	Latent	Generative	CUB	SUN	AWA	FLO	Average
AVSE (ours)	√	√	**61.5**	**63.6**	69.8	69.0	**66.0**
CVSE	√	×	54.8	57.5	56.3	49.5	54.5
AVFG	×	√	59.0	60.5	67.4	67.8	63.7
LATEM [33]	√	×	49.3	55.3	55.1	40.4	50.0
DEVISE [9]	√	×	52.0	56.5	54.2	45.9	52.2
ALE [0]	√	×	54.9	58.1	59.9	48.5	55.4
GAZSL [42]	×	√	55.8	61.3	68.2	60.5	61.5
f-CLSWGAN [39]	×	√	57.3	60.8	68.2	67.2	63.4
Cycle-CLSWGAN [8]	×	√	58.6	59.9	66.8	67.7	63.3
LisGAN [18]	×	√	58.8	61.7	**70.6**	**69.6**	65.2

Table 2: Compared results of zero-shot learning on four datasets. We report the top-1 accuracy (T) for unseen classes. The last column shows the average accuracy over the four datasets. *'Latent'* indicates the methods learn latent embeddings, while *'Generative'* means the methods need to generate features in an adversarial manner.

	CUB			SUN			AWA			FLO		
Method	U	S	H	U	S	H	U	S	H	U	S	H
AVSE (ours)	**51.5**	57.4	**54.3**	**47.4**	**39.5**	**43.1**	59.3	65.1	62.1	56.7	80.9	66.7
CVSE	22.6	56.3	32.2	20.4	33.5	25.3	16.3	74.5	26.7	12.4	59.2	20.5
AVFG	48.2	59.0	53.1	45.6	35.8	40.1	58.4	63.0	60.6	58.5	74.2	65.4
LATEM [33]	15.2	57.3	24.0	14.7	28.8	19.5	7.3	71.7	13.3	6.6	47.6	11.5
DEVISE [9]	23.8	53.0	32.8	16.9	27.4	20.9	13.4	68.7	22.4	9.9	44.2	16.2
ALE [0]	23.7	**62.8**	34.4	21.8	33.1	26.3	16.8	76.1	27.5	13.3	61.6	21.9
GAZSL [42]	23.9	60.6	34.3	21.7	34.5	26.7	19.2	**86.5**	31.4	28.1	77.4	41.2
f-CLSWGAN [39]	43.7	57.7	49.7	42.6	36.6	39.4	57.9	61.4	59.6	59.0	73.8	65.6
Cycle-CLSWGAN [8]	47.9	59.3	53.0	47.2	33.8	39.4	**59.6**	63.4	59.8	**61.6**	69.2	65.2
LisGAN [18]	46.5	57.9	51.6	42.9	37.8	40.2	52.6	76.3	**62.3**	57.7	**83.8**	**68.3**

Table 3: Compared results of generalized zero-shot learning. U and S measure the top-1 accuracy for unseen and seen classes, respectively. H is the harmonic mean of U and S.

forms better than CVSE and AVFG across the metrics, and obtains competitive performance with other methods. For example, our H accuracy is the highest on CUB and SUN (54.3% and 43.1%), the second highest on FLO (66.7%) and AWA (62.1%).

4.2 Ablation Study

We further implement several ablation variants for AVSE (Table 4), including **M1** by using only the adversarial loss $\mathcal{L}_{adv}$, **M2** by combining $\mathcal{L}_{adv}$ with the classification loss $\mathcal{L}_{cls}$, and **M3** that is a full model trained with all the three objectives. First, **M2** obtains a large improvement over **M1**, such as the T accuracy on CUB increasing from 56.8% to 60.1%. It reveals the importance of the classifier for learning class-distinct embedding. Second, compared to **M2**, adding the regression loss in **M3** has further improvements for both T and H accuracy. Third, in contrast to **M3**, we implement another full model **M4** without sharing the classifier C and the regressor R between visual and semantic embedding. However, the performance of **M4** is lower than that of **M3**. It implies that the sharing mechanism helps leveraging the relations between the two embeddings.

Model Variants	CUB		SUN		AWA		FLO	
	T	H	T	H	T	H	T	H
M1: AVSE ($\mathcal{L}_{adv}$)	56.8	46.9	58.1	34.4	66.4	57.9	65.2	63.2
M2: AVSE ($\mathcal{L}_{adv}+\mathcal{L}_{cls}$)	60.1	51.4	61.8	39.7	68.5	60.6	67.5	66.4
M3: AVSE ($\mathcal{L}_{adv}+\mathcal{L}_{cls}+\mathcal{L}_{reg}$)	**61.5**	**54.3**	**63.6**	**43.1**	**69.8**	**62.1**	**69.0**	**66.7**
M4: AVSE-unshared	60.7	52.8	62.4	41.3	69.0	61.4	68.4	66.1

Table 4: Ablation study on several AVSE variants. We report the top-1 accuracy T in the ZSL setting and the harmonic mean H in the GZSL setting.

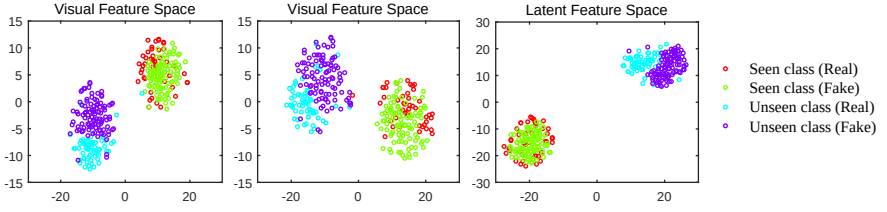


Figure 4: Analyzing the distributions of real and fake samples. Left: f-CLSWGAN [39]. Middle: Lis-GAN [18]. Right: Our AVSE. We select a seen class (*i.e.* Spotted Catbird) and an unseen class (*i.e.* American Pipit) from CUB. Each method synthesizes 100 fake samples for the two classes. AVSE separates the two classes with a larger distance.

4.3 Detailed Analysis

We provide more experiments to further analyze the strength of AVSE and compare it with two recent generative methods, f-CLSWGAN [39] and Lis-GAN [18].

Distribution of real and fake samples. Figure 4 visualizes the t-SNE distributions [31] for a seen class and an unseen class from CUB. For f-CLSWGAN and Lis-GAN, real samples represent the ground-truth visual features and fake samples are the synthetic visual features. For AVSE, visual and semantic embeddings in the latent space act as real and fake samples, respectively. We can see that AVSE leads to a large distance between the seen class and the unseen class, while f-CLSWGAN and Lis-GAN suffer from the misclassification due to the small distance between the two classes. Furthermore, we quantify the distributions with between-class distance and within-class variance. For any seen class, we compute its prototype by average the visual embeddings of its image instances. For each unseen class, its prototype is based on averaging a number of synthetic semantic embeddings. We calculate the between-class distance between any two prototypes, and the within-class variance by comparing the distance of each sample to its prototype. As reported in Table 5, AVSE can enlarge the between-class distance, as well as retain a small within-class variance.

Method	Dist.	Var.
f-CLSWGAN	0.568	0.136
Lis-GAN	0.588	0.129
AVSE	**0.654**	**0.122**

Table 5: Quantifying the distributions of CUB classes with between-class distance (Dist.) and within-class variance (Var.).

Number of synthetic samples. We analyze the effect of increasing the number of synthetic samples on the performance (*i.e.* T accuracy) of ZSL. The number we test ranges from 1 to 500 (Fig 5). It is worth noting that AVSE obtains good performance, even if we generate few samples per class. Specifically, when the number is 1, the accuracy for AVSE is 49.7%

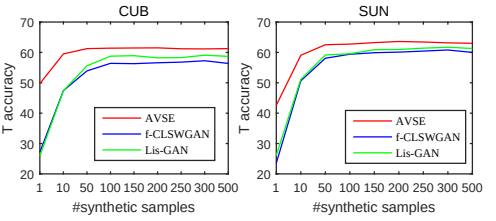


Figure 5: Analyzing the effect of the number of synthetic samples on the T accuracy in ZSL.

Unseen class	ZSL	GZSL
	**f-CLSWGAN:** White breasted Nuthatch; Loggerhead Shrike; Mockingbird; Groove billed Ani; Tree Swallow **Lis-GAN:** White breasted Nuthatch; Loggerhead Shrike; Tree Swallow; Groove billed Ani; Scott Oriole **AVSE:** Loggerhead Shrike; Groove billed Ani; Tree Swallow; Red legged Kittiwake; Yellow billed Cuckoo	**f-CLSWGAN:** Great Grey Shrike; White breasted Nuthatch; Loggerhead Shrike; Mockingbird; Black throated Blue Warbler **Lis-GAN:** Great Grey Shrike; White breasted Nuthatch; Loggerhead Shrike; Black throated Blue Warbler; Black throated Sparrow **AVSE:** Loggerhead Shrike; Red legged Kittiwake; Groove billed Ani; Tree Swallow; Forsters Tern
	**f-CLSWGAN:** watering hole; tundra; field cultivated; canal natural; bog **Lis-GAN:** watering hole; canal natural; bog; vineyard; field cultivated **AVSE:** bog; parking lot; watering hole; workshop; motel;	**f-CLSWGAN:** marsh; estuary; pond; lake natural; river **Lis-GAN:** watering hole; marsh; estuary; pond; bog **AVSE:** marsh; estuary; bog; parking lot; vineyard

Figure 6: Analyzing different predictions from ZSL and GZSL. The first unseen class is from CUB and the second one is from SUN. We show the class labels of the top-5 predictions under ZSL (Middle) and GZSL (Right). The seen class labels and unseen class labels are in green and blue, respectively. The ground-truth labels are underlined.

on CUB and 42.6% on SUN, which is largely higher than that of f-CLSWGAN (27.2% and 23.6%) and Lis-GAN (25.9% and 26.1%). When the number is larger than 100, the performance becomes stable. Without loss of generalization, we use AVSE to synthesize 200 samples per class for all the four datasets.

Qualitative comparison. Moreover, we exhibit some qualitative results in Fig. 6. AVSE estimates better predictions than f-CLSWGAN and Lis-GAN. In addition, we can observe the different predictions under the ZSL and GZSL settings. Since GZSL needs to classify both seen and unseen classes, its predictions are more difficult than ZSL.

5 Conclusion

We have proposed a simple and effective baseline model for zero-shot learning. The components in AVSE are pre-existing, however, integrating those components to design a generic baseline model is as important as exploring a specific approach. In addition, we perform embedding-to-image generation which visually exhibits the embeddings. The results across the datasets consistently show the improvements of our AVSE over previous baselines. AVSE has the potential to serve as a new baseline for ZSL. In the future,it is encouraged to improve the quality of synthetic data by imposing graph structure.

Acknowledgements. This research was funded by the FWO project “Structure from Semantics" (grant number G086617N).

References

[1] Zeynep Akata, Scott E. Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. Evaluation of output embeddings for fine-grained image classification. In *CVPR*, pages 2927–2936, 2015.

[2] Zeynep Akata, Florent Perronnin, Zaïd Harchaoui, and Cordelia Schmid. Label-embedding for image classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(7):1425–1438, 2016.

[3] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian D. Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, pages 3674–3683, 2018.

[4] Lei Jimmy Ba, Kevin Swersky, Sanja Fidler, and Ruslan Salakhutdinov. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *ICCV*, pages 4247–4255, 2015.

[5] Xiaojun Chang, Yi Yang, Alexander G. Hauptmann, Eric P. Xing, and Yaoliang Yu. Semantic concept discovery for large-scale zero-shot event detection. In *IJCAI*, pages 2234–2240, 2015.

[6] Soravit Changpinyo, Wei-Lun Chao, and Fei Sha. Predicting visual exemplars of unseen classes for zero-shot learning. In *ICCV*, pages 3496–3505, 2017.

[7] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, pages 2625–2634, 2015.

[8] Rafael Felix, B. G. Vijay Kumar, Ian D. Reid, and Gustavo Carneiro. Multi-modal cycle-consistent generalized zero-shot learning. In *ECCV*, pages 21–37, 2018.

[9] Andrea Frome, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *NIPS*, pages 2121–2129, 2013.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

[11] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. In *NIPS*, pages 5769–5779, 2017.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[13] Michael Kampffmeyer, Yinbo Chen, Xiaodan Liang, Hao Wang, Yujia Zhang, and Eric P Xing. Rethinking knowledge graph propagation for zero-shot learning. In *CVPR*, pages 11487–11496, 2019.

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[15] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[16] Elyor Kodirov, Tao Xiang, and Shaogang Gong. Semantic autoencoder for zero-shot learning. In *CVPR*, pages 4447–4456, 2017.

[17] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(3):453–465, 2014.

[18] Jingjing Li, Mengmeng Jing, Ke Lu, Zhengming Ding, Lei Zhu, and Zi Huang. Leveraging the invariant side of generative zero-shot learning. In *CVPR*, pages 21–37, 2018.

[19] Yujia Li, Kevin Swersky, and Richard S. Zemel. Generative moment matching networks. In *ICML*, pages 1718–1727, 2015.

[20] Ming Liang and Xiaolin Hu. Recurrent convolutional neural network for object recognition. In *CVPR*, pages 3367–3375, 2015.

[21] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[22] Yang Long and Ling Shao. Learning to recognise unseen classes by A few similes. In *ACM on Multimedia Conference*, pages 636–644, 2017.

[23] Yang Long, Li Liu, Fumin Shen, Ling Shao, and Xuelong Li. Zero-shot learning using synthesised unseen visual data with diffusion regularisation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(10):2498–2512, 2018.

[24] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[25] M-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008.

[26] Mark Palatucci, Dean Pomerleau, Geoffrey E. Hinton, and Tom M. Mitchell. Zero-shot learning with semantic output codes. In *NIPS*, pages 1410–1418, 2009.

[27] Genevieve Patterson and James Hays. SUN attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, pages 2751–2758, 2012.

[28] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. In *CVPR, DeepVision workshop*, 2014.

[29] Scott E. Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *CVPR*, pages 49–58, 2016.

[30] Mehdi S. M. Sajjadi, Giambattista Parascandolo, Arash Mehrjou, and Bernhard Schölkopf. Tempered adversarial networks. In *ICML*, pages 4448–4456, 2018.

[31] L.J.P van der Maaten and G.E. Hinton. Visualizing high-dimensional data using t-sne. *JMLR*, 9: 2579–2605, 2008.

[32] Vinay Kumar Verma, Gundeep Arora, Ashish Mishra, and Piyush Rai. Generalized zero-shot learning via synthesized examples. In *CVPR*, pages 4281–4289, 2018.

[33] 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.

[34] Qian Wang and Ke Chen. Zero-shot visual recognition via bidirectional latent embedding. *IJCV*, 124(3):356–383, 2017.

[35] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. *CVPR*, pages 6857–6866, 2018.

[36] Zheng Wang, Ruimin Hu, Chao Liang, Yi Yu, Junjun Jiang, Mang Ye, Jun Chen, and Qingming Leng. Zero-shot person re-identification via cross-view consistency. *IEEE Trans. Multimedia*, 18 (2):260–272, 2016.

[37] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. Zero-shot learning - a comprehensive evaluation of the good, the bad and the ugly. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(9):2251–2265, 2018.

[38] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh N. Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *CVPR*, pages 69–77, 2016.

[39] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *CVPR*, pages 5542–5551, 2018.

[40] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, pages 5908–5916, 2017.

[41] Haofeng Zhang, Yang Long, Yu Guan, and Ling Shao. Triple verification network for generalized zero-shot learning. *IEEE Trans. Image Process.*, 28(1):506–517, 2019.

[42] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. In *CVPR*, pages 3010–3019, 2017.

[43] Ziming Zhang and Venkatesh Saligrama. Zero-shot learning via joint latent similarity embedding. In *CVPR*, pages 6034–6042, 2016.

[44] Yizhe Zhu, Mohamed Elhoseiny, Bingchen Liu, Xi Peng, and Ahmed Elgammal. A generative adversarial approach for zero-shot learning from noisy texts. In *CVPR*, pages 1004–1013, 2018.