Learning Feature-to-Feature Translator by Alternating Back-Propagation for Generative Zero-Shot Learning

Yizhe Zhu[1]*, Jianwen Xie[2], Bingchen Liu[1], Ahmed Elgammal[1]
[1]Department of Computer Science, Rutgers University [2]Hikvision Research Institute
yizhe.zhu@rutgers.edu, jianwen@ucla.edu, bingchen.liu@rutgers.edu, elgammal@cs.rutgers.edu

Abstract

We investigate learning feature-to-feature translator networks by alternating back-propagation as a general-purpose solution to zero-shot learning (ZSL) problems. It is a generative model-based ZSL framework. In contrast to models based on generative adversarial networks (GAN) or variational autoencoders (VAE) that require auxiliary networks to assist the training, our model consists of a single conditional generator that maps class-level semantic features and Gaussian white noise vector accounting for instance-level latent factors to visual features, and is trained by maximum likelihood estimation. The training process is a simple yet effective alternating back-propagation process that iterates the following two steps: (i) the inferential back-propagation to infer the latent factors of each observed example, and (ii) the learning back-propagation to update the model parameters. We show that, with slight modifications, our model is capable of learning from incomplete visual features for ZSL. We conduct extensive comparisons with existing generative ZSL methods on five benchmarks, demonstrating the superiority of our method in not only ZSL performance but also convergence speed and computational cost. Specifically, our model outperforms the existing state-of-the-art methods by a remarkable margin up to 3.1% and 4.0% in ZSL and generalized ZSL settings, respectively.

1. Introduction

Deep learning techniques have successfully tackled various computer vision problems such as object detection [16, 34, 10, 26, 9, 63], image, video and 3d shape generation [18, 13, 57, 58, 33, 60, 59, 64, 45], pose estimation [32, 44, 65, 42, 43], object recognition [20, 41, 17], etc. Especially, these advanced deep learning methods equip the machine with the comparable ability of object recognition to human beings when abundant labeled training samples are provided. However, this ability will decrease dramatically for classes that are insufficiently represented or even not present in the training data, thus increasing the difficulty of real-world applications due to the costly data collection and annotation effort. This limitation attracts the intense interest of researchers in zero-shot learning (ZSL).

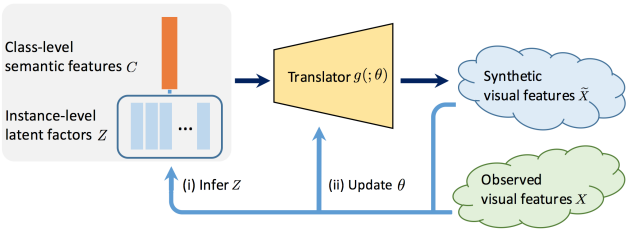


Figure 1: Demonstration of the feature-to-feature translator for generative zero-shot learning. Training (blue flow): given the class-level semantic features C and the observed visual features X, (i) the latent factors Z are inferred by MCMC and (ii) the weights θ of the translator are updated by the gradient ascent for maximum likelihood. Translation (black flow): once the translator is learned, arbitrary number of synthetic visual features can be generated for unseen classes by translating the unseen class semantic features along with Z randomly sampled from Gaussian distribution. The generated samples for unseen classes are useful for zero-shot learning.

ZSL aims to recognize novel classes where no training data is available for these classes. The key to make ZSL work is to use the semantic description of classes as the bridge to connect the seen classes and unseen classes. Recently, generative ZSL approaches [4, 67, 55, 11, 50, 46] have emerged as a new trend of ZSL strategy by exploiting the successful generative models, e.g., variational autoencoders (VAE) [18, 35], generative adversarial networks (GAN) [13, 28], etc, to learn mappings from class-level semantic features (e.g., attributes or description embeddings) to visual features. The synthetic visual features for unseen classes through well-trained generative models establish the visual description of unseen classes and make the conven-

*Work was done while Yizhe Zhu was an intern at Hikvision.

tional supervised classification applicable. The quality of generative ZSL approaches mainly depends on how well the generative model can emulate the data distribution. GAN-based methods [67, 55, 11] employ the conditional generator network which maps class-level semantic features along with Gaussian white noise as latent factors to visual features. The conditional generator is trained in an adversarial manner where a well-designed discriminator network is recruited to play a minimax game with the conditional generator. The adversarial training strategy gets around the intractable inference of latent factors in training; however, the imbalance between the discriminator and the generator would lead to non-convergence and mode collapse issues. VAE-based methods [50, 46] associate the conditional generator network with an additional encoder that approximates the posterior distribution for the purpose of inference of latent factors, and train both models by maximizing the variational lower bound. In this paper, we will show that neither of these assisting networks is necessary for training the conditional generator.

In our framework of generative ZSL, we adopt a conditional generator as a feature-to-feature translator network that translates the class-level semantic features along with the instance-level latent factors to visual features as shown in Figure 1. However, different from GAN and VAE-based methods, we resort to a theoretically more accurate estimator, maximum likelihood estimation (MLE) with EM-like strategy [6], to train the network, without the requirement of auxiliary networks for assistance. The hard nut to crack in training latent variable models by MLE is the intractable posterior distribution that is required for computing the gradient of the observed-data log-likelihood. In the proposed framework, we adopt Markov chain Monte Carlo (MCMC), such as Langevin dynamics [51, 29], for computing the posterior distribution. We show that the maximum likelihood algorithm involves the computation of the gradient of observed-data log-likelihood with respect to model parameters and the gradient with respect to latent factors, both of which can be efficiently computed by back-propagation.

Specifically, the proposed translator is learned by alternating back-propagation (ABP) algorithm, which iterates the following two steps: (i) *inferential back-propagation*: for each training example, inferring the continuous latent factors by sampling from the current learned posterior distribution via Langevin dynamics, where the gradient of the log joint density can be calculated by back-propagation, (ii) *learning back-propagation*: updating the model parameters given the inferred latent factors and the training examples by gradient ascent, where the gradient of the log-likelihood with respect to the model parameters can again be calculated by back-propagation.

The ABP algorithm was originally proposed for training unconditional generator networks [15, 56]. In this paper, we generalize ABP to learning conditional generator network for feature-to-feature translation, where the class-level semantic features play the role of condition.

It is worth mentioning that the proposed model is capable of learning from incomplete training examples, where visual features are partially corrupted or inaccessible due to occlusion in image space. Specifically, our model can learn from incomplete data by making latent factors only explain the visible parts of the visual features conditioned on class-level semantic features, while GAN or VAE-based methods can hardly deal with this situation.

Our contributions can be summarized as follows: (1) We propose a feature-to-feature translator for generative ZSL, where we learn a mapping from class-level semantic features, along with instance-level latent factors following Gaussian white noise distribution, to visual features. (2) We propose to learn the translator via alternating back-propagation (ABP) algorithm for maximum likelihood, without relying on other assisting networks, which makes our framework more statistically rigorous and elegant. (3) We show that the proposed framework can learn from incomplete training examples where visual features are partially visible due to corruption or occlusion in image space. (4) Comprehensive experiments conducted on various ZSL tasks show the state-of-the-art performance of our framework, and a thorough analysis of the model demonstrates the superiority of the proposed framework in different aspects.

2. Related Work

X-to-Y Translation. We are not the first to apply conditional generators to learn X-to-Y mapping. A text-to-image translation is proposed for image synthesis from text description [33]. Zhu *et al* [66] has studied image-to-image translation problem for different types of image processing tasks, which include synthesizing photos from label maps or edge maps and generating color images from their grey-scaled versions. Recently, video-to-video translation problem has been tackled by learning a mapping function from an input source video (e.g., a sequence of semantic segmentation masks) to a target realistic video [49]. Our work learns a feature-to-feature mapping for ZSL. Additionally, all other works mentioned above are based on the framework of GANs, which means that well-designed discriminator networks need to be resorted to in the training stage. Our framework differs in that it is trained by an alternating back-propagation algorithm without incorporating any extra assisting networks. This makes our framework considerably simpler and computationally more efficient than those based on GANs.

Generative Models. Our model is essentially a conditional latent variable model. The alternating back-propagation (ABP) training method for our model is re-

lated to variational inference (e.g., VAE) and adversarial learning (e.g., GAN), both of which require an extra assisting network with a separate set of learning parameters to avoid the explaining-away inference of latent variables in the model. Unlike VAE and GAN, ABP does not involve an auxiliary network and performs explicit explaining-away inference by directly sampling from the posterior distribution via MCMC, such as Langevin dynamics, which is powered by back-propagation. Our model trained by ABP is much simpler, more natural and statistically rigorous than those trained by adversarial learning and variational inference schemes. ABP has been used to train general generators [15] and deformable generators [61] for image patterns, as well as dynamic generators [56] for video patterns. Our paper is a generalization of [15, 56] by applying ABP to train a conditional version of the generator model for feature-to-feature translation. The generative ConvNet [58] and the Wasserstein INN [23] are two one-piece models that learn energy-based generative models for data generation. Both [58] and [23] generate data via iterative MCMC sampling, while our model generates data via direct ancestral sampling, which is much more efficient.

Zero-Shot Learning. Several pioneering works for ZSL [21, 22] make use of class attributes as intermediate information to classifiy images for unseen classes. Some ZSL methods are based on the bilinear compatibility function between the visual and semantic features, which can be learned by using (a) the ranking loss (e.g., ALE [1] and DeViSE [12]), (b) the structural SVM loss [2] or (c) the ridge regression loss (e.g., ESZSL [36] and PTZSL [8]). To enhance the expressive power of the models, several ZSL approaches [53, 62, 25, 24, 68] learn non-linear multimodal embedding. Taking advantage of the generative models in data generation, several methods [67, 11, 52, 46, 4, 50] resort to generating visual features from unseen classes for ZSL. Both GAZSL [67] and FGZSL [52] pair a Wasserstein GAN [3, 14] with a classification loss as regularization to increase the inter-class discrimination of synthetic features. MCGZSL [11] adopts cycle consistency loss [66] to regularize the generator for ZSL . [50, 46] employ conditional VAEs [39] framework to learn feature generator. Our model also learns to generate visual features from unseen classes by a conditional generator, however, different from the generative ZSL methods mentioned above, our model is trained by alternating back-propagation, without the need of assisting models for training.

3. Feature-to-Feature Translator

3.1. Conditional Latent Variable Model for Feature-to-Feature Translation

Let $S = \{(X_i, C_i), i = 1, ..., n\}$ be the training data of the seen classes, where $X_i \in \mathcal{X}^D$ is the D-dimensional visual features (e.g., CNN features extracted from images) for the i-th image and $C_i \in \mathcal{C}^K$ is its K-dimensional class-level semantic features (e.g., class attribute embedding). Because one class usually corresponds to many image examples, we aim at finding a one-to-many class-to-instance feature generator. Specifically, we try to learn a mapping $g : \mathcal{C}^K \times \mathcal{Z}^d \rightarrow \mathcal{X}^D$ that seeks to explain the visual features X_i extracted from each image by its corresponding class-level features C_i and a d-dimensional vector of latent factors $Z_i \in \mathcal{Z}^d$ that accounts for instance variations. We assume Z_i is sampled from a Gaussian prior distribution $N(0, I_d)$, where I_d stands for the d-dimensional identity matrix. Once the generator g learns to generate image features from class features, it can also generate $\widetilde{X}$ from any unseen classes. Formally, the feature-to-feature mapping can be formulated by a conditional latent variable model as follows:

$$\begin{aligned} &Z \sim \mathcal{N}(0, I_d), \\ &X = g_\theta(C, Z) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2 I_D), \end{aligned} \tag{1}$$

where θ contains all the learning parameters in the mapping function g, and ϵ is a D-dimensional noise vector following Gaussian distribution. The mapping g can be any non-linear mapping. In this paper we adopt the top-down MLP parameterization of g, which is also called the conditional generator network, to map the latent factors Z along with the class features C to the visual features X. Generally, the model is defined by a prior distribution of latent factors $Z \sim p(Z)$ and the conditional distribution to generate visual features given the class features and the latent factors, i.e., $[X|Z, C] \sim p_\theta(X|Z, C)$. Let $q_{\text{data}}(X|C)$ be the true distribution that generates the training visual features given their associated class features. The goal of learning this model is to minimize the Kullback-Leibler divergence $\text{KL}(q_{\text{data}}(X|C) \| p_\theta(X|C))$ over θ.

3.2. Learning by Alternating Back-Propagation

3.2.1 Maximum Likelihood Learning

The maximum likelihood estimation (MLE) of our model $p_\theta(X|C)$ is equivalent to minimizing the Kullback-Leibler divergence $\text{KL}(q_{\text{data}}(X|C) \| p_\theta(X|C))$ over θ. The complete data model is given by

$$\begin{aligned} &\log p_\theta(X, Z|C) = \log[p_\theta(X|Z, C)p(Z)] \\ &= -\frac{1}{2\sigma^2}\|X - g_\theta(C, Z)\|^2 - \frac{1}{2}\|Z\|^2 + \text{const}, \end{aligned} \tag{2}$$

where the constant term is independent of X, Z and θ. The observed-data model or the log-likelihood is obtained by integrating out the latent factors Z:

$$\log p_\theta(X|C) = \log \int p_\theta(X|Z, C)p(Z)dZ. \tag{3}$$

Suppose we observe the training data $S = \{(X_i, C_i), i = 1, ..., n\}$, and $[X|C] \sim p_\theta(X|C)$, the goal of MLE is to maximize the observed-data log-likelihood:

$$L(\theta) = \sum_{i=1}^{n} \log p_\theta(X_i|C_i). \tag{4}$$

Without loss of generality, we consider one observed pair (X_i, C_i) and omit subscript i for brevity. To maximize the log-likelihood, we adopt gradient ascent strategy. The gradient of the log-likelihood with respect to θ can be calculated by the following equation:

$$\begin{aligned} \frac{\partial}{\partial \theta} \log p_\theta(X|C) &= \frac{1}{p_\theta(X|C)} \frac{\partial}{\partial \theta} p_\theta(X|C) \\ &= \mathbb{E}_{Z \sim p_\theta(Z|X,C)} \left[\frac{\partial}{\partial \theta} \log p_\theta(X, Z|C) \right]. \end{aligned} \tag{5}$$

3.2.2 Inferential Back-Popagation

Since the expectation with respect to the posterior distribution $p_\theta(Z|X, C)$ in Equation (5) is analytically intractable, we resort to the average of the MCMC samples to approximate the expectation. Specifically, we employ Langevin dynamics which carries out sampling by iterating:

$$\begin{aligned} Z_{\tau+1} &= Z_\tau + \frac{s^2}{2} \frac{\partial}{\partial Z} \log p_\theta(X, Z_\tau|C) + sU_\tau, \\ &= Z_\tau + sU_\tau + \\ &\frac{s^2}{2} \left[\frac{1}{\sigma^2}(X - g_\theta(C, Z_\tau)) \frac{\partial}{\partial Z} g_\theta(C, Z_\tau) - Z_\tau \right], \end{aligned} \tag{6}$$

where τ denotes the time step for Langevin dynamics, U_τ is the Gaussian white noise corresponding to the Brownian motion, which is added to prevent the chain from being trapped by local modes, and s is the step size.

Because of the high computational cost of MCMC, it is infeasible to generate independent samples from scratch in each learning iteration. In practice, the MCMC transition of latent factors Z in the current iteration starts from the previous updated result of Z, which is obtained from the previous learning iteration. We initialize Z with Gaussian white noise at the beginning. Persistent MCMC update with such a warm start scheme is effective and efficient enough to provide fair samples from the posterior distribution.

We infer the latent factors Z_i for each observed pair (X_i, C_i) by sampling a single copy of Z_i from $p_\theta(Z_i|X_i, C_i)$ via running finite steps of Langevin dynamics starting from the current Z_i (i.e., the warm start). This is a conditional explaining-away inference solving an inverse problem, which is that given the specific visual features of an image and its associated class features, how to obtain its corresponding latent factors. Unlike VAE, our model does not need to recruit an extra network for inference.

3.2.3 Learning Back-Popagation

Once Z is inferred, we learn the model via stochastic gradient algorithm by updating θ based on the Monte Carlo approximation of the gradient of $L(\theta)$ in Equation (5):

$$\theta_{t+1} = \theta_t + \gamma_t \frac{\partial}{\partial \theta} L(\theta), \tag{7}$$

where t is the time step for the gradient algorithm, γ_t is the learning rate, and

$$\begin{aligned} \frac{\partial}{\partial \theta} L(\theta) &\approx \sum_{i=1}^{n} \frac{\partial}{\partial \theta} \log p_\theta(X_i, Z_i|C_i) \\ &= -\sum_{i=1}^{n} \frac{\partial}{\partial \theta} \frac{1}{2\sigma^2} \|X_i - g_\theta(C_i, Z_i)\|^2 \\ &= \sum_{i=1}^{n} \frac{1}{\sigma^2}(X_i - g_\theta(C_i, Z_i)) \frac{\partial}{\partial \theta} g_\theta(C_i, Z_i). \end{aligned} \tag{8}$$

The Equation (7) corresponds to a non-linear regression problem, where it seeks to find the θ to predict visual features X_i by its corresponding observed class features C_i and the inferred latent factors Z_i.

3.2.4 Alternating Back-Propagation Algorithm

The key to compute Equation (6) is to calculate $\partial g_\theta(C, Z)/\partial Z$, while the key to compute Equation (8) is to calculate $\partial g_\theta(C, Z)/\partial \theta$. Both of them can be efficiently computed by back-propagation. Algorithm 1 describes the details of the learning and sampling algorithm.

Algorithm 1 Alternating back-propagation procedure for learning feature-to-feature translator.

Input: training samples $\{(X_i, C_i), i = 1, ..., n\}$, the maximal number of loops N_{step}, the number of Langevin steps l, the learning rate γ_t.

Output: the learned parameters θ, the inferred latent factors $\{Z_i, i = 1, ..., n\}$.

1: Initialize θ and Z_i, for $i = 1, ..., n$.
2: **for** t = 1,..., N_{step} **do**
3: **Inferential back-propagation**: For each i, run l steps of Langevin dynamics to sample $Z_i \sim p_\theta(Z_i|X_i, C_i)$ with warm start, i.e., starting from the current Z_i, each step follows equation (6).
4: **Learning back-propagation**: Update $\theta \leftarrow \theta + \gamma_t L'(\theta)$, where $L'(\theta)$ is computed according to equation (8), with learning rate γ_t.
5: **end for**

3.3. Comparison with Variational Inference

Our learning algorithm presented in Algorithm 1 seeks to minimize $\mathrm{KL}(q_{\text{data}}(X|C)\|p_\theta(X|C))$, while variational auto-encoder (VAE) changes the objective to

$$\min_{\theta,\phi} \mathrm{KL}(q_{\text{data}}(X|C)p_\phi(Z|X, C)\|p(Z|C)p_\theta(X|Z, C))$$

by utilizing an extra inference model $p_\phi(Z|X,C)$ with parameter ϕ to infer latent variable Z, where the inference model $p_\phi(Z|X,C)$ is an analytically tractable approximation to $p_\theta(Z|X,C)$. Compared to the maximum likelihood objective $\mathrm{KL}(q_{\text{data}}(X|C)\|p_\theta(X|C))$, which is the KL-divergence between the marginal distributions of X conditioned on C, while the VAE objective is the KL-divergence between the joint distributions of (Z,X) conditioned on C (i.e., an upper bound of the maximum likelihood objective.) as shown below:

$$\mathrm{KL}(q_{\text{data}}(X|C)p_\phi(Z|X,C)\|p_\theta(Z,X|C)) =$$
$$\mathrm{KL}(q_{\text{data}}(X|C)\|p_\theta(X|C)) + \mathrm{KL}(p_\phi(Z|X,C)\|p_\theta(Z|X,C)).$$

The accuracy of variational inference depends on the accuracy of the inference model $p_\phi(Z|X,C)$ as an approximation of the true posterior distribution $p_\theta(Z|X,C)$. That is, when $\mathrm{KL}(p_\phi(Z|X,C)\|p_\theta(Z|X,C)) = 0$, the variational inference is equivalent to the maximum likelihood solution. Therefore, our learning algorithm is more natural, straightforward, accurate, and computationally efficient than VAE.

4. Zero-Shot Learning

4.1. Classification in Zero-Shot Learning

The feature-to-feature translator that maps $[C,Z] \rightarrow X$ can be considered as an explicit implementation of the local linear embedding [37], where $[C,Z]$ is the embedding of X, with disentanglement of class features C and non-class features Z. Given any unseen class $C^u \in \mathcal{C}^K$, we can generate arbitrarily many visual features for the unseen class by first sampling Z from $\mathcal{N}(0,I_d)$ and then mapping (C^u,Z) into X via the learned feature-to-feature translator $\widetilde{X} = g_\theta(C^u,Z)+\epsilon$. With generated data of unseen classes, the labels of testing examples from unseen classes can be predicted via any conventional supervised classifiers. Here we employ a KNN classifier (K=20) for ZSL for its simplicity and a softmax classifier for GZSL as suggested in [55].

4.2. Learning from Incomplete Visual Features

The inferential back-propagation step performs an explaining-away inference, where the latent factors compete with each other to explain the observed visual features. This is very useful in the scenario where the training visual features are incomplete. In this case, our model is still able to learn from incomplete visual features via alternating back-propagation algorithm. The latent factors can still be obtained by explaining the incomplete observed visual features, and the model parameters can still be updated as before. Taking the features in [67] as an example, Zhu *et al* train a part detector to localize small semantic parts of birds such as heads and tails, and concatenate the extracted features of parts as the visual representation of the object for ZSL. The part features are inaccessible in two cases: (a) the semantic parts don't appear in the images, for example, no tails can be observed when birds are in front view; (b) the detector fails to discover the parts. In these cases, zeros are assigned to the corresponding bins of feature vector for the missing parts. We can easily adapt our ABP algorithm to the above situation by changing the computation of $\|X-g_\theta(C,Z)\|^2$ to $\|M\circ(X-g_\theta(C,Z))\|^2$, where M is the given binary indicator matrix with the same size of X, with 1 indicating "visible" and 0 indicating "missing", and sign $\circ$ denotes element-wise matrix multiplication operation. The indicator matrices M_i vary for different visual features X_i. Note that GAN or VAE-based methods can hardly handle learning from incomplete visual features.

5. Experiments

5.1. Experiment Settings

Datasets. We evaluate the proposed framework for generative ZSL on four widely used ZSL benchmark datasets and compare it with a number of state-of-the-art baselines. The datasets include: (1) Caltech-UCSD Birds-200-2011 (CUB) [48], (2) Animal with Attributes (AwA1) [22], (3) Animal with Attributes 2 (AwA2) [54], (4) SUN attribute (SUN) [31]. CUB is a fine-grained dataset of bird species with 312 class-level attributes. AwA1 is a coarse-grained dataset including 50 classes of animals with 85 attributes. Since the original images in AwA1 are not publicly available due to the copyright license issue, Xian *et al* [54] create a new dataset AwA2 by collecting new images for each class in AwA1 while keeping the attribute annotations the same as AwA1. SUN contains 717 types of scenes with 102 attributes. We follow the train/test split settings in [54], which ensures that no unseen classes are included in the ImageNet dataset where the visual feature extractor is pretrained, to avoid violating the setting of ZSL. Besides, we also conduct experiments on a large-scale dataset, i.e., ImageNet-21K [7]. In this challenging dataset, no attribute annotations are available. We use the word embedding of the class names as the semantic represention of the classes [5].

Implementation Details. Our translator is implemented by a multilayer perceptron with a single hidden layer of 4,096 nodes. LeakyReLU and ReLU are used as nonlinear activation functions on the hidden layer and the output layer respectively. The dimension of latent factors z is set to be 10. We fix σ = 0.3, l = 10 and s = 0.3. Our model is trained with a batch size of 64 and the Adam optimizer with a learning rate of 10^{-3}, β_1 = 0.9, and β_2 = 0.999. The number of epochs is 50. Our model is implemented using Pytorch framework [30] and trained on one NVIDIA TITAN Xp GPU. Our code is publicly available online[1] .

[1] https://github.com/EthanZhu90/ZSL_ABP

	Method	CUB	AwA1	AwA2	SUN
§	DAP [22]	40.0	44.1	46.1	39.9
	CMT [38]	34.6	39.5	37.9	39.9
	LATEM [53]	49.3	55.1	55.8	55.3
	ALE [1]	54.9	59.9	62.5	58.1
	DEVISE [12]	52.0	54.2	59.7	56.5
	SJE [2]	53.9	65.6	61.9	53.7
	ESZSL [36]	53.9	58.2	58.6	54.5
	SYNC [5]	55.6	54.0	46.6	56.3
	SAE [19]	33.3	53.0	54,1	40.3
	DEM [62]	51.7	65.7	66.5	60.8
	GFZSL [47]	49.3	68.3	63.8	60.6
†	VZSL [50]	56.3	<u>67.1</u>	66.8	59.0
	GAZSL [67]	55.8	63.7	64.2	<u>60.1</u>
	FGZSL [55]	57.7	65.6	66.9	58.6
	MCGZSL [11]	<u>58.4</u>	66.8	<u>67.3</u>	60.0
	Ours	**58.5**	**69.3**	**70.4**	**61.5**

Table 1: Performance comparison for zero-shot learning on CUB, AwA1, AwA2 and SUN datasets. The performance is measured by average per-class top-1 accuracy (%). † and § indicate generative and non-generative methods, respectively. The best and the second best results are marked in bold and underlined respectively.

5.2. Zero-Shot Learning

As for zero-shot learning setting, we follow the evaluation protocol used in [54], where results are measured by average per-class top-1 accuracy. We compare with 15 state-of-the-art methods, including 11 non-generative ones and 4 generative ones which are separately shown in Table 1. For GAZSL, FGZSL, and MCGZSL, we get results by running their codes. We reimplement the VZSL by ourselves since no code is publicly available. The results of other methods are published in [54]. From Table 1, we can observe that: (i) the generative methods have an overall performance superior to the non-generative ones (ii) our proposed model consistently outperforms previous state-of-the-art methods and shows great superiorities on AwA1 and AwA2, where the improvements are up to 3.1%. This validates that our model trained by alternating back-propagation is significantly beneficial to ZSL tasks.

5.3. Generalized Zero-Shot Learning

In the generalized zero-shot learning setting, a test image is classified to the union of seen and unseen classes. This setting is more practical and difficult as it removes the assumption that test images only come from unseen classes. Following the protocol proposed by [54], we compute the harmonic mean of accuracies on seen and unseen classes: $H = \frac{2 \cdot A_{\mathcal{S}} \cdot A_{\mathcal{U}}}{A_{\mathcal{S}} + A_{\mathcal{U}}}$, where $A_{\mathcal{S}}$ and $A_{\mathcal{U}}$ denote the accuracies of classifying images from seen classes and those from unseen classes respectively. We evaluate our method on four datasets and show the performance comparison with 13 state-of-the-art methods in Table 2. The experimental results show that non-generative methods achieving high performance on seen classes perform badly on unseen classes, indicating that those methods are biased in favor of seen classes. In contrast, those generative methods can mitigate the bias and perform well on both unseen and seen classes. Compared with other generative ZSL methods, our method obtains much higher accuracies on unseen classes and comparable accuracies on seen classes. It improves the state-of-the-art performances by notable margins on most datasets (e.g., 4.0% on AwA2 in terms of harmonic mean), demonstrating the capability for generalized zero-shot learning.

5.4. Large-Scale Experiments

We also evaluate the performance of our model on the large-scale ImageNet-21K [7] dataset. The dataset contains a total of 14 million images from more than 21K classes. The relation among classes follows the WordNet [27] hierarchy. Following the same protocol in [54, 5], we keep a specific subset of 1K classes for training, and use either all the remaining classes or a subset of it for testing. Specifically, the subsets for testing are determined according to the hierarchical distance from the training classes, or their population. For example, *2Hop* contains 1,509 unseen classes that are within two tree hops of the seen 1K classes based on the class hierarchy, while *3Hop* increases the number of unseen classes to 7,678 by extending the range to three tree hops. *M500*, *M1K* and *M5K* contain 500, 1K, and 5K most populated classes, while *L500*, *L1K* and *L5K* contain 500, 1K, and 5K least populated classes respectively.

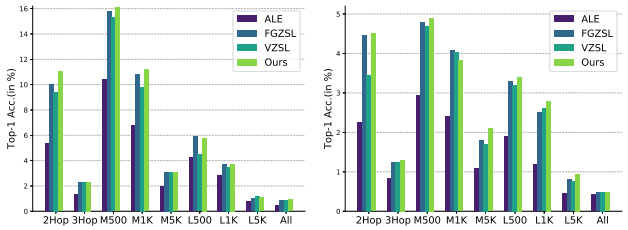
We compare our method with three baseline methods, which include a visual-semantic embedding-based method, i.e., ALE, and two generative model-based methods, i.e., VZSL and FGZSL. The results are shown in Figure 2. A notable observation is that all methods perform much better in *2Hop* subset than in *3Hop* subset. Two vital factors accounting for the observation are: in *3Hop* subset, (i) the unseen classes are semantically less related to the training classes, making it difficult to transfer knowledge, (ii) a dramatic increase in the number of unseen classes (from $1,509$ to $7,678$) makes the classification even harder. Generally, it is evident that generative methods have superior performances. Among the three generative methods, our proposed method achieves the best accuracies in most cases, demonstrating that it is very competitive in this realistic and challenging task.

5.5. Comparison with GAN & VAE-based Methods

To thoroughly evaluate the performance of our model, we perform extensive ablation experiments. We study the comparison of our method with GAN and VAE-based methods (i.e., FGZSL and VZSL) in terms of **(i)** the speed of

	Method	CUB			AwA1			AwA2			SUN		
		$A_\mathcal{U}$	$A_\mathcal{S}$	H	$A_\mathcal{U}$	$A_\mathcal{S}$	H	$A_\mathcal{U}$	$A_\mathcal{S}$	H	$A_\mathcal{U}$	$A_\mathcal{S}$	H
§	DAP [22]	1.7	**67.9**	3.3	0.0	**88.7**	0.0	0.0	84.7	0.0	4.2	25.1	7.2
	DEVISE [12]	23.8	53.0	32.8	13.4	68.7	22.4	17.1	74.7	27.8	16.9	27.4	20.9
	CMT [38]	7.2	49.8	12.6	0.9	87.6	1.8	0.5	**90.0**	1.0	8.1	21.8	11.8
	SJE [2]	23.5	59.2	33.6	11.3	74.6	19.6	8.0	73.9	14.4	14.7	30.5	19.8
	LATEM [53]	15.2	57.3	24.0	7.3	71.7	13.3	11.5	77.3	20.0	14.7	28.8	19.5
	ESZSL [36]	12.6	63.8	21.0	6.6	75.6	12.1	5.9	77.8	11.0	11.0	27.9	15.8
	ALE [1]	23.7	62.8	34.4	16.8	76.1	27.5	14.0	81.8	23.9	21.8	33.1	26.3
	SAE [19]	7.8	54.0	13.6	1.8	77.1	3.5	1.1	82.2	2.2	8.8	18.0	11.8
	DEM [62]	19.6	57.9	29.2	32.8	84.7	47.3	30.5	86.4	45.1	20.5	34.3	25.6
†	VZSL [50]	44.9	54.1	49.1	53.4	68.3	59.9	51.7	67.2	58.4	43.5	34.9	38.7
	GAZSL [67]	26.5	57.4	36.2	32.8	84.7	47.3	**59.9**	68.3	53.4	21.7	34.5	26.7
	FGZSL [19]	45.9	54.6	49.9	53.1	68.0	59.6	50.2	67.5	57.5	40.2	36.4	38.2
	MCGZSL [11]	45.7	61.0	**52.3**	56.9	64.0	60.2	51.9	67.2	58.6	**49.4**	33.6	40.0
	Ours	**47.0**	54.8	50.6	**57.3**	67.1	**61.8**	55.3	72.6	**62.6**	45.3	**36.8**	**40.6**

Table 2: Performance comparison for generalized zero-shot learning. † and § indicate generative and non-generative model-based methods respectively. The best and the second best results are marked in bold and underlined respectively.



(a) ZSL (b) GZSL

Figure 2: ZSL and GZSL results on ImageNet. For GZSL, $A_\mathcal{U}$ is reported.

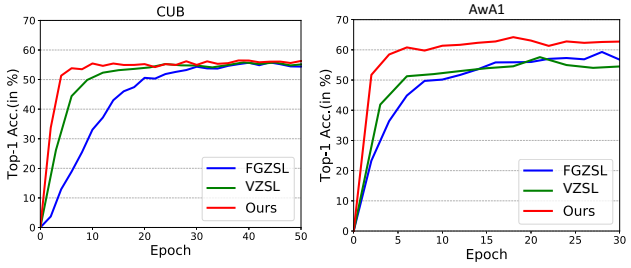


Figure 3: Convergence comparison: top-1 accuracies in the validation set over different numbers of training epochs.

the convergence, **(ii)** the number of model parameters, **(iii)** ZSL performance when generating different numbers of features for unseen classes, **(iv)** ZSL performance under different numbers of seen classes for training.

Comparison (i) To investigate the advantage of our proposed method, we display the convergence curves of different models. We reserve a subset of seen classes as the validation set, and the remaining seen classes are used for training. Considering that various loss functions used in different methods are not comparable, we instead use the average per-class top-1 accuracy of the validation set as the convergence indicator. Figure 3 shows classification accuracies over different numbers of training epochs on CUB and AwA1 datasets. The convergence trends on both datasets are similar. Among these three methods, ours converges fastest, while the GAN-based method is the slowest one.

Dataset	# of Parameters	# of Mult-Adds
FGZSL [55]	20.62M	41.23M
VZSL [50]	21.90M	43.78M
Ours	9.71M	19.42M

Table 3: Comparison of the number of parameters and the computational cost among three generative model-based ZSL methods that are applied to CUB dataset.

Comparison (ii) Another advantage of our proposed model is the small number of parameters, or equivalently, the low computational load. We compare our method with FGZSL and VZSL for the CUB dataset in terms of the number of parameters and the number of element-wise multiplication and addition operations in Table 3. Note that these numbers might get changed when the methods are applied to other datasets, because different datasets may have various dimensions of semantic and visual features. Due to the incorporation of auxiliary networks in VAE and GAN-based methods (i.e., the encoder in VAE and the discriminator in GAN), these models require more parameters and computations. Our method only contains one single conditional generator, therefore its parameter size and computational cost are only half of those in GAN and VAE-based frameworks.

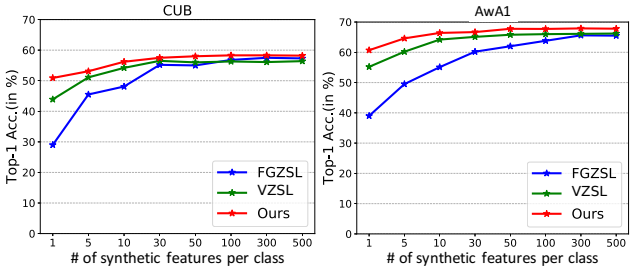


Figure 4: Comparison of top-1 per-class accuracies of unseen classes with different numbers of synthetic features per class.

Comparison (iii) Figure 4 shows the accuracies of three methods with different numbers of synthetic features for each unseen class. The same curve trends appear on both CUB and AwA1 datasets. The results are better with larger number of synthetic samples. When the number increases to 300, all the performances are similar and stable, indicating that each of the models is saturated as no improvement appears with more samples. Compared with FGZSL, our model performs much better with fewer samples (e.g., 50.1% v.s. 29.1% on CUB and 60.1% v.s. 38.9% on AwA1 with only 1 sample). The VZSL model performs slightly worse than ours.

Comparison (iv) As pointed out in [40], the number of unseen classes usually dramatically surpasses that of seen classes in the real world. However, most ZSL benchmark datasets are far away from this situation. For instance, only 72 out of 717 classes on the SUN dataset are specified as unseen classes. This motivates us to investigate how models perform with different numbers of seen classes for training. We conduct experiments on the SUN dataset as it contains more classes than other datasets. We keep the unseen classes the same and randomly sample different numbers of seen classes for training. To make the experiments closer to the real world situation, we report the performance in the generalized ZSL setting, as shown in Figure 5. As the number of seen classes increases, the accuracies of seen classes of all methods consistently decrease, indicating the increasing difficulty in discriminating seen class examples. From another perspective, more seen classes can provide more knowledge to associate the visual and semantic features, resulting in the improvement in the performance on unseen classes. Compared with other generative methods, our model achieves the best $A_{\mathcal{U}}$ while keeping decent $A_{\mathcal{S}}$.

5.6. Learning from Incomplete Visual Features

Zhu *et al* [8, 67] propose to use concatenated part features for ZSL, where those part features are either from groundtruth part annotations or extracted by a learned part detector, denoted as GTA and DET features respectively. The missing ratio in GTA is 9.3%, while the ratio decreases

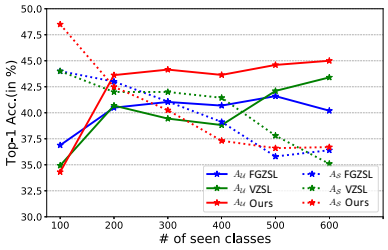


Figure 5: Comparison of top-1 per-class accuracies of unseen classes among different methods with different numbers of seen classes in training.

			DET*			
Method	GTA	DET	30%	50%	70%	90%
GAZSL [67]	74.1	72.7	68.5	63.7	55.6	37.7
Ours	76.7	75.2	72.9	71.3	64.8	51.6

Table 4: Zero-shot learning performance (top-1 accuracy %) of the models trained on incomplete visual features with different missing ratios.

to 4.0% in DET due to the high recall of their part detector. We first evaluate the performance of our model and GAZSL [67] with GTA features and DET features. As shown in Table 4, our method outperforms GAZSL by 2.6% and 2.5% using GTA and DET features respectively. To further analyze the power of our model in dealing with incomplete visual features, we increase the missing ratio of the DET features by randomly masking some valid feature values. As the missing ratio increases, the performances of both methods drop due to the reason that less and less useful information can be used. However, our method can still achieve a decent performance of 51.6% in the most challenging situation where the missing ratio reaches 90%, while the accuracy of GAZSL is only 37.7%. This confirms the claimed power of our model in learning from incomplete visual features for ZSL.

6. Conclusion

We propose a feature-to-feature translator learned by an alternating back-propagation algorithm as a general-purpose solution to zero-shot learning. Unlike other generative models, such as GAN and VAE, our method is simple yet effective, and does not rely on any assisting networks for training. The alternating back-propagation algorithm iterates the inferential back-propagation for inferring the instance-level latent factors and the learning back-propagation for updating the model parameters. We present a solution to learning from incomplete visual features for ZSL. We show that our framework outperforms the existing generative ZSL methods.

Acknowledgment. This work is partially supported by NSFUSA award 1409683. We thank Prof. Ying Nian Wu for helpful discussions.

References

[1] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for image classification. *TPAMI*, 38(7):1425–1438, 2016. 3, 6, 7

[2] Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. Evaluation of output embeddings for fine-grained image classification. In *CVPR*, pages 2927–2936, 2015. 3, 6, 7

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, pages 214–223, 2017. 3

[4] Maxime Bucher, Stéphane Herbin, and Frédéric Jurie. Generating visual representations for zero-shot classification. In *ICCV Workshops*, 2017. 1, 3

[5] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Synthesized classifiers for zero-shot learning. In *CVPR*, pages 5327–5336, 2016. 5, 6

[6] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (methodological)*, pages 1–38, 1977. 2

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 5, 6

[8] Mohamed Elhoseiny, Yizhe Zhu, Han Zhang, and Ahmed Elgammal. Link the head to the "beak": Zero shot learning from noisy text description at part precision. In *CVPR*, pages 6288–6297, 2017. 3, 8

[9] Deng-Ping Fan, Ming-Ming Cheng, Jiang-Jiang Liu, Shang-Hua Gao, Qibin Hou, and Ali Borji. Salient objects in clutter: Bringing salient object detection to the foreground. In *ECCV*, pages 186–202, 2018. 1

[10] Deng-Ping Fan, Wenguan Wang, Ming-Ming Cheng, and Jianbing Shen. Shifting more attention to video salient object detection. In *CVPR*, pages 8554–8564, 2019. 1

[11] Rafael Felix, Vijay BG Kumar, Ian Reid, and Gustavo Carneiro. Multi-modal cycle-consistent generalized zero-shot learning. In *ECCV*, pages 21–37, 2018. 1, 2, 3, 6, 7

[12] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. Devise: A deep visual-semantic embedding model. In *NIPS*, pages 2121–2129, 2013. 3, 6, 7

[13] 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. 1

[14] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of Wasserstein GANs. In *NIPS*, pages 5767–5777, 2017. 3

[15] Tian Han, Yang Lu, Song-Chun Zhu, and Ying Nian Wu. Alternating back-propagation for generator network. In *AAAI*, 2017. 2, 3

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, pages 2961–2969, 2017. 1

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1

[19] Elyor Kodirov, Tao Xiang, and Shaogang Gong. Semantic autoencoder for zero-shot learning. In *CVPR*, pages 3174–3183, 2017. 6, 7

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 1

[21] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, pages 951–958, 2009. 3

[22] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *TPAMI*, 36(3):453–465, 2014. 3, 5, 6, 7

[23] Kwonjoon Lee, Weijian Xu, Fan Fan, and Zhuowen Tu. Wasserstein introspective neural networks. In *CVPR*, pages 3702–3711, 2018. 3

[24] Kai Li, Martin Renqiang Min, Bing Bai, Yun Fu, and Hans Peter Graf. On novel object recognition: A unified framework for discriminability and adaptability. In *ACM International Conference on Information and Knowledge Management*, 2019. 3

[25] Kai Li, Martin Renqiang Min, and Yun Fu. Rethinking zero-shot learning: A conditional visual classification perspective. In *ICCV*, 2019. 3

[26] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *ECCV*, pages 21–37, 2016. 1

[27] George A Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995. 6

[28] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 1

[29] Radford M Neal et al. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11):2, 2011. 2

[30] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS Workshops*, 2017. 5

[31] Genevieve Patterson and James Hays. SUN attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, pages 2751–2758, 2012. 5

[32] Xi Peng, Zhiqiang Tang, Fei Yang, Rogerio S Feris, and Dimitris Metaxas. Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation. In *CVPR*, pages 2226–2234, 2018. 1

[33] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *ICML*, pages 1060–1069, 2016. 1, 2

[34] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015. 1

[35] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, pages 1278–1286, 2014. 1

[36] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, pages 2152–2161, 2015. 3, 6, 7

[37] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. 5

[38] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *NIPS*, pages 935–943, 2013. 6, 7

[39] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *NIPS*, pages 3483–3491, 2015. 3

[40] Jie Song, Chengchao Shen, Yezhou Yang, Yang Liu, and Mingli Song. Transductive unbiased embedding for zero-shot learning. In *CVPR*, pages 1024–1033, 2018. 8

[41] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 1

[42] Zhiqiang Tang, Xi Peng, Shijie Geng, Lingfei Wu, Shaoting Zhang, and Dimitris Metaxas. Quantized densely connected U-nets for efficient landmark localization. In *ECCV*, pages 339–354, 2018. 1

[43] Zhiqiang Tang, Xi Peng, Shijie Geng, Yizhe Zhu, and Dimitris Metaxas. CU-net: Coupled U-nets. *BMVC*, 2018. 1

[44] Zhiqiang Tang, Xi Peng, Tingfeng Li, Yizhe Zhu, and Dimitris Metaxas. Adatransform: Adaptive data transformation. In *ICCV*, 2019. 1

[45] Yu Tian, Xi Peng, Long Zhao, Shaoting Zhang, and Dimitris N Metaxas. CR-GAN: learning complete representations for multi-view generation. In *IJCAI*, pages 942–948, 2018. 1

[46] Vinay Kumar Verma, Gundeep Arora, Ashish Mishra, and Piyush Rai. Generalized zero-shot learning via synthesized examples. In *CVPR*, pages 4281–4289, 2018. 1, 2, 3

[47] Vinay Kumar Verma and Piyush Rai. A simple exponential family framework for zero-shot learning. In *ECML-PKDD*, pages 792–808, 2017. 6

[48] 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. 5

[49] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *NIPS*, pages 1152–1164, 2018. 2

[50] Wenlin Wang, Yunchen Pu, Vinay Kumar Verma, Kai Fan, Yizhe Zhang, Changyou Chen, Piyush Rai, and Lawrence Carin. Zero-shot learning via class-conditioned deep generative models. *AAAI*, 2018. 1, 2, 3, 6, 7

[51] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *ICML*, pages 681–688, 2011. 2

[52] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, pages 499–515, 2016. 3

[53] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *CVPR*, pages 69–77, 2016. 3, 6, 7

[54] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learninga comprehensive evaluation of the good, the bad and the ugly. *TPAMI*, 41(9):2251–2265, 2019. 5, 6

[55] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *CVPR*, pages 5542–5551, 2018. 1, 2, 5, 6, 7

[56] Jianwen Xie, Ruiqi Gao, Zilong Zheng, Song-Chun Zhu, and Ying Nian Wu. Learning dynamic generator model by alternating back-propagation through time. In *AAAI*, 2019. 2, 3

[57] Jianwen Xie, Yang Lu, Ruiqi Gao, Song-Chun Zhu, and Ying Nian Wu. Cooperative training of descriptor and generator networks. *arXiv preprint arXiv:1609.09408*, 2016. 1

[58] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. A theory of generative convnet. In *ICML*, pages 2635–2644, 2016. 1, 3

[59] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, and Ying Nian Wu. Learning descriptor networks for 3D shape synthesis and analysis. In *CVPR*, pages 8629–8638, 2018. 1

[60] Jianwen Xie, Song-Chun Zhu, and Ying Nian Wu. Synthesizing dynamic patterns by spatial-temporal generative convnet. In *CVPR*, pages 7093–7101, 2017. 1

[61] Xianglei Xing, Tian Han, Ruiqi Gao, Song-Chun Zhu, and Ying Nian Wu. Unsupervised disentangling of appearance and geometry by deformable generator network. In *CVPR*, pages 10354–10363, 2019. 3

[62] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. In *CVPR*, pages 2021–2030, 2017. 3, 6, 7

[63] Jia-Xing Zhao, Yang Cao, Deng-Ping Fan, Ming-Ming Cheng, Xuan-Yi Li, and Le Zhang. Contrast prior and fluid pyramid integration for RGBD salient object detection. In *CVPR*, pages 3927–3936, 2019. 1

[64] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris Metaxas. Learning to forecast and refine residual motion for image-to-video generation. In *ECCV*, pages 387–403, 2018. 1

[65] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for 3D human pose regression. In *CVPR*, pages 3425–3435, 2019. 1

[66] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pages 2223–2232, 2017. 2, 3

[67] 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. 1, 2, 3, 5, 6, 7, 8

[68] Yizhe Zhu, Jianwen Xie, Zhiqiang Tang, Xi Peng, and Ahmed Elgammal. Learning where to look: Semantic-guided multi-attention localization for zero-shot learning. *arXiv preprint arXiv:1903.00502*, 2019. 3