Gradient Matching Generative Networks for Zero-Shot Learning

Mert Bulent Sariyildiz
Bilkent University
Department of Computer Engineering
mert.sariyildiz@bilkent.edu.tr

Ramazan Gokberk Cinbis
Middle East Technical University (METU)
Department of Computer Engineering
gcinbis@metu.edu.tr

Abstract

Zero-shot learning (ZSL) is one of the most promising problems where substantial progress can potentially be achieved through unsupervised learning, due to distributional differences between supervised and zero-shot classes. For this reason, several works investigate the incorporation of discriminative domain adaptation techniques into ZSL, which, however, lead to modest improvements in ZSL accuracy. In contrast, we propose a generative model that can naturally learn from unsupervised examples, and synthesize training examples for unseen classes purely based on their class embeddings, and therefore, reduce the zero-shot learning problem into a supervised classification task. The proposed approach consists of two important components: (i) a conditional Generative Adversarial Network that learns to produce samples that mimic the characteristics of unsupervised data examples, and (ii) the Gradient Matching (GM) loss that measures the quality of the gradient signal obtained from the synthesized examples. Using our GM loss formulation, we enforce the generator to produce examples from which accurate classifiers can be trained. Experimental results on several ZSL benchmark datasets show that our approach leads to significant improvements over the state of the art in generalized zero-shot classification.

1. Introduction

There has been tremendous progress in visual recognition models over the past several years, primarily driven by the advances in deep learning. The state-of-the-art approaches in deep learning, however, predominantly rely on the availability of a large set of carefully annotated training examples. The need for such large-scale datasets poses a significant bottleneck against building comprehensive recognition models of the visual world, especially due to the long-tailed distribution of object categories [1].

Recently, there has been a significant research interest in overcoming this difficulty. Prominent approaches for this purpose include semi-supervised learning, *i.e*. improving supervised classification through leveraging unlabeled data [2, 3], few-shot learning, *i.e*. learning from few labeled samples [4, 5] and zero-shot learning (ZSL) for modeling novel classes without training samples [6, 7, 8]. In our paper, we focus on the ZSL problem, where the goal is to extrapolate a classification model learned from *seen classes*, *i.e*. those with labeled examples, to *unseen classes* with no labeled training samples. In order to relate classes to each other, they are commonly represented as *class embedding* vectors constructed from side information. Such class embedding vectors can be constructed in several different ways, such as by manually defining attributes that characterize visual and semantic properties of objects [9, 10, 11] or by adapting vector-space embeddings of class names [12, 13, 14] or by representing the position of classes in a relevant taxonomy tree as vectors [15]. Given the class embeddings, the ZSL problem boils down to modeling relations between a visual feature space, *i.e*. images or features extracted from some deep convolutional network, and a class embedding space [16, 15, 17, 18, 19, 20, 21, 22].

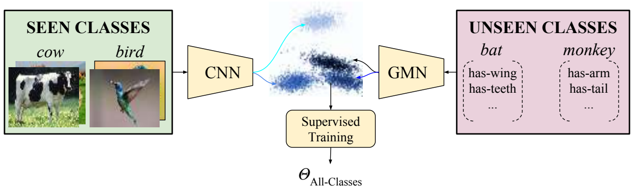

Figure 1: Illustration of our approach. We propose the Gradient Matching Network (GMN) which learns to produce synthetic examples for a class given it's semantic embedding. By using the GMN we generate training samples for zero-shot (*unseen*) classes, then train a supervised classifier over the union of this synthetic set and the training set of *seen* classes.

However, ZSL models typically suffer from the domain shift problem [23] due to distributional differences between seen and unseen classes. This can significantly limit the *generalized zero-shot learning* (GZSL) accuracy where test samples may belong to any of the seen or unseen classes [24]. Towards addressing this problem, several recent work have proposed generative models that can synthesize training samples for unseen classes and learn a classifier from real and/or synthesized examples [25, 26, 27, 28]. Therefore, a bias towards seen classes can be reduced considerably.

Similarly, in this work, our goal is to learn a generative model that can synthesize samples for any class of interest, purely based on the embedding vector of the class. Once the generative model is learned, we augment the set of seen class examples by the set of unseen class examples sampled from the generative model. The final classification model is then built by training a classifier over the real and synthetic training examples, Therefore, in a sense, we reduce ZSL to a supervised learning problem, as illustrated in Fig 1.

Just like any other example-synthesis based ZSL approach, however, the accuracy of the resulting classifier heavily depends on the diversity and fidelity of the training examples synthesized by the generative model. For this reason, we specifically focus on two directions: (i) leveraging unseen examples to implicitly model the manifold of each unseen class, (ii) ensuring that generative model produces data using which we can train an accurate classifier.

In order to leverage unlabeled examples, we propose as a Generative Adversarial Network (GAN) [29] based formulation. In particular, in contrast to recent GAN-based example synthesis approaches [26, 27, 28], our approach allows utilizing an unconditional GAN discriminator, which naturally extends to incorporating over unlabeled training examples. In this way, we aim to learn a generator that produces samples that mimic the characteristics of both seen and unseen classes.

In order to learn to generate better training data, we propose a novel loss function that behaves as a quality inspector on the synthetic samples. More specifically, we aim to guide the generator towards minimizing the classification loss of synthetic example-driven classification models. For this purpose, we derive the *gradient matching loss* as a proxy for the classification loss, which measures the discrepancy between the gradient vectors obtained using the real versus synthetic samples. We refer to our complete model that incorporates this loss term as *Gradient Matching Network* (GMN).

We show that our final classification models lead to state-of-the-art results on ZSL benchmark datasets Caltech-UCSD Birds (CUB) [30], SUN Attributes (SUN) [31] and Animals with Attributes (AWA) [11] using the challenging and realistic Generalized ZSL (GZSL) evaluation protocols [24, 32].

The rest of the paper is organized as follows: Section 2 provides an overview of the most relevant previous work, Section 3 explains the details of the proposed approach, and Section 4 presents empirical evaluations of the method. Finally, Section 5 concludes the paper with a brief discussion.

2. Related Work

In this section, we provide an overview of the most related work on zero-shot learning.

Over the years, a number of ZSL approaches have been proposed. For example, [6] models the joint probability of attributes, [8] models the conditional distribution of features given attributes, [33] uses semantic knowledge bases for attribute classification, [34, 35, 36] build convex combinations of seen class classifiers, [15, 17, 18, 19, 20, 21, 22] learn a compatibility function between features and class embeddings. Similarly, [37, 38, 39] learn a mapping from semantic embeddings to visual features, and, [40, 41, 42] learn a data-driven metric for comparing similarities between features and semantic embeddings. Alternatively, transductive approaches have been proposed to benefit from unlabeled data [43, 23, 44, 39]. Such discriminative techniques, however, typically assume that each unlabeled example belongs to one of the unseen (or seen) classes, which can be an unrealistic assumption in practice.

Recently, the use of contemporary generative models in zero-shot learning settings has gained attention. [45] proposes training a conditional Variational Auto-Encoder (cVAE), that learns to generate samples according to given class embeddings. [44] extends this notion with trainable class conditional latent spaces. [28] also develops a cVAE except that their model learns a separate semantic embedding regressor/discriminator. [25] evaluates several generative models for learning to generate training examples. [27] adopts cycle consistency loss of cycle-GAN into zero-shot learning to regularize feature synthesis network. [46] uses a separate reconstructor, discriminator and classifier all targeting at visual features to remedy domain-shift problem. Slightly different from mainstream approaches, [47] introduces diffusion regularization to increase utility of features. [26] proposes a WGAN [48] based formulation that uses a discriminative supervised loss function, in addition to the unsupervised adversarial loss. In this model, the supervised loss enforces the WGAN generator to produce samples that are correctly classified according to a pre-trained classifier of seen classes.

Among the aforementioned works, [26] is the one closest to ours in the sense that we also train a conditional WGAN towards synthesizing training samples. However, our approach has two major differences. First, we use the proposed gradient matching loss, which aims to directly maximize the value of the produced training examples by measuring the quality of the gradient signal obtained over the synthesized examples. Second, our model learns an unconditional discriminator *i.e.*, the discriminator network does not rely on a semantic embedding vector. This permits us to explore the incorporation of unlabeled training examples into training in a semi-supervised fashion.

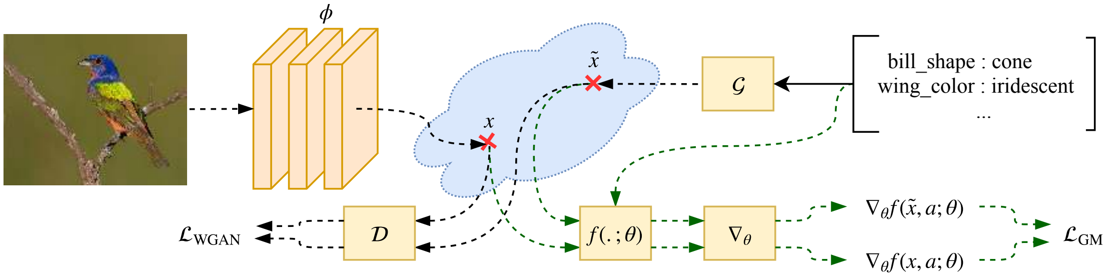
ϕ
$\tilde{x}$
x
$\mathcal{G}$
bill_shape : cone
wing_color : iridescent
...
$\mathcal{L}_{\text{WGAN}}$
$\mathcal{D}$
$f(.;\theta)$
∇_θ
$\nabla_\theta f(\tilde{x}, a; \theta)$
$\nabla_\theta f(x, a; \theta)$
$\mathcal{L}_{\text{GM}}$

Figure 2: Illustration of the gradient matching loss. ϕ is a pre-trained CNN. $\mathcal{G}$ is the generator and it synthesizes features for any class using its semantic embedding. $\mathcal{D}$ represents the discriminator network. f is the compatibility function. ∇ denotes gradient operator in the compute graph. Paths through which only data of seen classes flow when $\mathcal{D}$ is unconditional are colored in green. (Best viewed in color.)

3. Method

In ZSL, the goal is to learn a classifier on a set of *seen* classes for which we have training samples, and then to use this function to predict the class labels of test samples belonging to *unseen* classes for which we have no training data. In addition to the conventional ZSL, in GZSL, the test samples may also belong to the seen classes. To enable knowledge transfer to novel classes, one can define an auxiliary (semantic embedding) space $\mathcal{A}$, in which both seen and unseen classes can be uniquely identified. This way, the classifier can be formulated as a compatibility function $f(x, a; \theta_f) : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$, which estimates the degree of confidence that the input image (or its representation) $x \in \mathcal{X}$ belongs to the class represented by the embedding $a \in \mathcal{A}$, using the model with parameters θ_f. Given the compatibility function, the classifier over all classes can be constructed.

We start by defining a set of seen classes $\mathcal{Y}_{\text{s}} = \{1, \ldots, C_s\}$ and a set of unseen classes $\mathcal{Y}_{\text{u}} = \{C_s + 1, \ldots, C_s + C_u\}$ such that $\mathcal{Y}_{\text{s}} \cap \mathcal{Y}_{\text{u}} = \emptyset$ and $\mathcal{Y}_{\text{all}} = \mathcal{Y}_{\text{s}} \cup \mathcal{Y}_{\text{u}}$. For each class in $\mathcal{Y}_{\text{all}}$, there is a unique class embedding vector $a \in \mathbb{R}^{d_a}$, and we denote the set of all class embeddings by $\mathcal{A}_{\text{all}}$. Thus $\mathcal{A}_{\text{s}}$ and $\mathcal{A}_{\text{u}}$ represent embeddings of seen and unseen classes, respectively. $\mathcal{D}_{\text{train}} = \{(x, a) \mid x \in \mathcal{X}_{\text{s}}, a \in \mathcal{A}_{\text{s}}\}$, is the training set containing N examples, where each training example consists of the feature representation $x \in \mathbb{R}^{d_x}$ extracted using a pre-trained CNN, and the corresponding class embedding vector a. Here, $\mathcal{X}_{\text{s}}$ denotes the set of all labeled data points. During training, our approach can optionally utilize a set of unlabeled examples, denoted by $\mathcal{X}_{\text{u}}$.

3.1. Unsupervised GAN

Our generative model is built upon the WGAN [49] as in [26]. Different from vanilla GAN [29], WGAN optimizes the Wasserstein distance using Kantorovich Rubinstein duality, instead of optimizing the Jensen-Shannon divergence. It is shown that enforcing discriminators to be 1-Lipschitz provides more stable gradients for generators. Even though clipping the weights of discriminators serves this purpose, it leads to unstable training for the WGAN. Instead, [48] propose to apply gradient penalty to discriminators to control their Lipschitz norm, which we use as our starting point:

$$\mathcal{L}_{\text{WGAN}} = \mathop{\mathbb{E}}_{x \sim P_r} [\mathcal{D}(x)] - \mathop{\mathbb{E}}_{\tilde{x} \sim P_g} [\mathcal{D}(\tilde{x})] + \lambda \mathop{\mathbb{E}}_{\hat{x} \sim P_{\hat{x}}} \left[(\|\nabla_{\hat{x}} \mathcal{D}(\hat{x})\|_2 - 1)^2 \right], \tag{1}$$

where, P_r is the true data distribution, P_g denotes generator outputs and $\hat{x}$ is the interpolation between x and $\tilde{x}$.

Note that Eq. 1 does not involve any label information regarding either the real samples from the data distribution $x \sim P_r$ or fake ones synthesized by the generator $\tilde{x} \sim P_g$. In order to generate a sample $\tilde{x}$, a noise vector shall be sampled from a prior distribution and then fed into the generator in a purely unsupervised manner. In our case, however, we aim to produce training samples for the unseen classes using the generative model. For this purpose, we need to train a generator network such that it takes a combination of the noise vector and class embeddings as input, and therefore, produces class-specific samples according to the side information given by the class embedding.

A simple scheme for combining the noise and class embedding vectors is to concatenate them [26]. However, we can also aim to model the latent distributions corresponding to classes and then take samples from these latent distributions instead. For this purpose, inspired from [44], we propose to define a conditional multivariate Gaussian distribution $\mathcal{N}(\mu(a), \Sigma(a))$, where $\mu(a) = \mathcal{W}_{\text{mu}} a + b_{\text{mu}}$ and $\Sigma(a) = \exp(\mathcal{W}_{\text{cov}} a + b_{\text{cov}})$ estimate an $\mathbb{R}^{d_z}$ dimensional Gaussian noise mean and covariance conditioned on class embeddings, $\mathcal{W}_{\text{mu}}$ and $\mathcal{W}_{\text{cov}}$ are linear transformation matrices and b_{mu} and b_{cov} are bias vectors. Therefore, in order to generate a sample of class j, we first compute $\mu(a_j)$ and $\Sigma(a_j)$, take a noise sample from $\mathcal{N}(\mu(a), \Sigma(a))$ and then feed the noise into the generator network. To make the sampling process differentiable, we use the re-parameterization trick [50, 51]. In this manner, we make $\mathcal{W}_{\text{mu}}$, $\mathcal{W}_{\text{cov}}$, b_{mu} and b_{cov} end-to-end differentiable and train them as an integral part of the generative network.

3.2. Gradient Matching Loss

So far the aforementioned approach lacks definition of any supervisory signal, which is crucial for learning a correct conditional generative model (See the *Ablation Study* in Sec. 4). One possible solution is to measure the correctness of the resulting samples for the seen classes using the loss function of a pre-trained classification model, which is the approach used in [26]. However, we argue that classification guidance does not necessarily lead to the synthesis of a good training set as it measures the loss of the samples w.r.t. the pre-trained model, rather than the expected loss of the model trained on them. For instance, if the generator learns to generate only confidently classified examples, the classification loss given by the pre-trained model will be low, even though the resulting training set lacks examples near class boundaries, *i.e*. the support vectors. In fact, [52, 53] report that conditional GAN models tend to produce degenerate class conditional examples when they are trained to minimize the loss of a pre-trained classifier.

Based on these observations we propose that instead of aiming to produce samples that are correctly classified by a pre-trained model, we should focus on learning to generate training examples that lead to accurate classification models. For this purpose, one can consider training the generative model by minimizing the final loss of a tentative classification model trained over the synthetic samples. Here, the tentative classifier would be iteratively trained via a gradient based optimizer over a number of model update steps, within each training iteration for the generative model. Since all computational blocks are differentiable, such an approach would allow training the generative model in an end-to-end fashion such that it learns to generate training examples from which accurate classification models can be built.

However, based on our preliminary experiments, we have observed that such a naive strategy performs poorly for two important reasons. First, normally a large number of model update steps are needed to be able to train the tentative classifier. However, integrating a long compute chain of model update steps within the generative model training procedure not only slows down the training procedure very significantly, but also leads to vanishing gradient problems. Second, using an unrealistically small number of classifier update steps due to the aforementioned problems, on the other hand, encourages the generative model to produce unrealistic samples that aim to "quickly" minimize the final loss over the few classification model update steps.

Instead, we address these issues by focusing on maximizing the correctness of individual model updates. We observe the simple fact that in the case where a generative model learns true class manifolds, partial derivatives of a loss function with respect to classification model parameters over a large set of synthetic examples would be highly correlated with those over a large set of real training examples.

Following these observations, we propose to minimize the approximation error of the gradients obtained over the synthetic samples of seen classes. More specifically, we propose to learn a generative model $\mathcal{G}$ such that it maximizes the correlation between gradients over the synthetic samples and those over the real samples. To formalize this idea, we first define the aliases g_{r} and g_{s} for the expected gradient vectors over the real and synthesized examples, respectively:

$$g_{\text{r}}(\theta) = \mathop{\mathbb{E}}_{(x,a)\sim\mathcal{D}_{\text{s}}} \left[\nabla_{\theta_f} \mathcal{L}_{\text{CLS}}(f, x, a; \theta_f = \theta)\right], \tag{2}$$

$$g_{\text{s}}(\theta) = \mathop{\mathbb{E}}_{\tilde{x}\sim\mathcal{G}(a\sim\mathcal{A}_{\text{s}})} \left[\nabla_{\theta_f} \mathcal{L}_{\text{CLS}}(f, \tilde{x}, a; \theta_f = \theta)\right]. \tag{3}$$

Here, $\mathcal{L}_{\text{CLS}}(f, x, a)$ is the loss function used in training the compatibility function $f(x, a; \theta_f)$. Throughout the training procedure, we approximate g_{r} and g_{s} over sample batches.

Since the most important information conveyed by the gradient vector is the direction towards the local minima rather than the absolute scale of the gradient vectors, we measure the discrepancy between g_{r} and g_{s} via the cosine similarity between two vectors. Finally, we formalize the *gradient matching loss* $\mathcal{L}_{\text{GM}}$ as the expected cosine distance between the g_{r} and g_{s}, computed over all possible compatibility model parameters θ:

$$\mathcal{L}_{\text{GM}} = \mathop{\mathbb{E}}_{\theta} \left[1 - \frac{g_{\text{r}}(\theta)^{\text{T}} g_{\text{s}}(\theta)}{\|g_{\text{r}}(\theta)\|_2 \|g_{\text{s}}(\theta)\|_2}\right], \tag{4}$$

In our experiments, we approximate the expectation by sampling θ_f vectors obtained over the training iterations while learning the compatibility function via gradient descent over real training examples. Then our final objective becomes

$$\theta^*_{\mathcal{G}}, \theta^*_{\mathcal{D}} = \arg\min_{\theta_{\mathcal{G}}, \theta_{\mathcal{D}}} \{\mathcal{L}_{\text{WGAN}} + \beta\mathcal{L}_{\text{GM}}\}, \tag{5}$$

where β is a simple weight hyper-parameter to be tuned on a validation set. We refer to a generative model trained within this framework as Gradient Matching Network (GMN).

Given the true generative model of the data distribution and a representative train set, the correlation between $g_{\text{r}}(\theta)$ and $g_{\text{s}}(\theta)$ is expected to be high, independent of the compatibility model parameters θ. Therefore, in principle, any compatibility function model can be utilized within the gradient matching loss. In our experiments, we use cross-entropy as $\mathcal{L}_{\text{CLS}}$ and implement the compatibility function f as a bilinear model:

$$f(x, a; W, b) = x^{\text{T}} W a + b. \tag{6}$$

The compatibility matrix W and bias vector b corresponds to θ. We note that while optimizing $\mathcal{L}_{\text{GM}}$ by a batch gradient descent update rule, it is important to compute $g_{\text{r}}(\theta)$ and $g_{\text{s}}(\theta)$ over real and synthetic samples of the same class, respectively. This makes sure that the genarator effectively learns class manifolds separately. Otherwise, although matching the aggregated gradients ∇_{θ_f} of a batch of samples that belong to different classes is still a valid supervision for the generator to learn the data distribution, it becomes difficult for the generator to learn individual class distributions.

Furthermore, thanks to our gradient matching loss, we can decouple the class label supervision from $\mathcal{L}_{\text{WGAN}}$ objective. This way, depending on the availability of unlabeled training data, $\mathcal{L}_{\text{WGAN}}$ term in Eq. 5 can be computed either over seen class embeddings and samples ($\mathcal{L}^{\text{S}}_{\text{WGAN}}$), or, over all classes ($\mathcal{L}^{\text{S+U}}_{\text{WGAN}}$) possibly in a transductive way:

$$\mathcal{L}^{\text{S}}_{\text{WGAN}} = \mathop{\mathbb{E}}_{\tilde{x}\sim\mathcal{G}(a\sim\mathcal{A}_{\text{s}})}[\mathcal{D}(\tilde{x})] - \mathop{\mathbb{E}}_{x\sim\mathcal{X}_{\text{s}}}[\mathcal{D}(x)] + \lambda\mathcal{L}_{\text{GP}} \tag{7}$$

$$\mathcal{L}^{\text{S+U}}_{\text{WGAN}} = \mathop{\mathbb{E}}_{\tilde{x}\sim\mathcal{G}(a\sim\mathcal{A}_{\text{all}})}[\mathcal{D}(\tilde{x})] - \mathop{\mathbb{E}}_{x\sim\mathcal{X}_{\text{all}}}[\mathcal{D}(x)] + \lambda\mathcal{L}_{\text{GP}}, \tag{8}$$

where $\mathcal{L}_{\text{GP}}$ is the gradient penalty term in Eq. 1. Here, in the case of Eq. 8, $\mathcal{D}_{\text{train}}$ also includes $\mathcal{X}_{\text{u}}$. Unlike most of the transductive zero-shot learning approaches, we do not assume that unseen examples belong solely to the unseen classes: while such an assumption can possibly provide a significant advantage in training, it is unrealistic in most scenarios. The compute graph summarizing our approach is depicted in Fig. 2.

3.3. Supervision by Conditional Discriminator

Up to now, the source of supervision for a generator network is defined by an auxiliary loss function minimized by the generator network itself during training. However, we can also condition a discriminator network on either one-hot class labels or semantic embedding vectors so that it can also learn relations between visual features and semantic embeddings [26, 27, 28, 46]. To do that we slightly change Eq. 1 as follows:

$$\mathcal{L}^{\text{S}}_{\text{cWGAN}} = \mathbb{E}\left[\mathcal{D}(x,a)\right] - \mathbb{E}\left[\mathcal{D}(\tilde{x},a)\right] + \lambda\,\mathbb{E}\left[(\|\nabla_{\hat{x}}\mathcal{D}(\hat{x},a)\|_2 - 1)^2\right]. \tag{9}$$

We note that this conditional form of the discriminator network can only be trained using training samples of seen classes. In other words, it cannot be utilized over unsupervised samples in a semi-supervised or transductive settings.

In our experiments, we comprehensively evaluate the impact of training with different GAN loss versions ($\mathcal{L}^{\text{S}}_{\text{WGAN}}$, $\mathcal{L}^{\text{S+U}}_{\text{WGAN}}$, $\mathcal{L}^{\text{S}}_{\text{cWGAN}}$) and their combinations with gradient matching loss.

3.4. Feature Synthesis

Once we train our generative model, we synthesize training examples for both seen and unseen classes by providing their class embeddings into the generative network as input, and then we combine the resulting $\mathcal{D}_{\text{fake}}$ with $\mathcal{D}_{\text{train}}$ to form our final training set $\tilde{\mathcal{D}} = D_{\text{train}} \cup \mathcal{D}_{\text{fake}}$. Once all samples are generated, we train the multi-class classification model based on the compatibility function by simply minimizing the cross-entropy loss over all (seen+unseen) classes. Finally, we utilize the resulting f to perform ZSL and GZSL.

4. Experiments

In this section, we present an experimental evaluation of the proposed approach. First we briefly explain our experimental setup, then we evaluate important GMN variants and compare with the state of the art. We additionally analyze our model via a detailed ablation study, including an evaluation on the effect of using synthesized training examples.

	n_{attr}	$\lvert\mathcal{Y}_u\rvert$	$\lvert\mathcal{Y}_s\rvert$	$\lvert\mathcal{X}_u\rvert$	$\lvert\mathcal{X}_s\rvert$	*ANOSPC*
CUB [30]	312	50	150	2967	8821	59
SUN [31]	102	72	645	1440	12900	20
AWA [11]	85	10	40	5685	24790	609

Table 1: Statistics for the benchmark datasets. n_{attr} denotes number of attributes and $|.|$ indicates cardinality of a set. The last column shows *average number of samples per class* over each of the datasets. We use the splits proposed in [32].

Datasets. We evaluate our model in the three commonly used benchmark datasets, namely Caltech-UCSD Birds-200-2011 (CUB) [30], SUN Attribute (SUN) [31] and Animals with Attributes (AWA) [11]. CUB and SUN are fine-grained image datasets which contain 200 bird species and 717 scene categories, respectively. They are particularly challenging for ZSL & GZSL as they contain relatively fewer images per class, making it difficult to model intra-class variations efficiently. AWA is a coarse-grained dataset consisting of images belonging to 50 animals. AWA contains a relatively small set of classes, which makes generalization to unseen classes more difficult. A summary is given in Table 1.

In our comparisons, we utilize the splits, class embeddings and evaluation metrics proposed in [32] for standardized ZSL and GZSL evaluation. We use class-level attributes as class embeddings. For CUB experiments, we additionally use 1024-dimensional character-based CNN-RNN features [54] as in [32, 27]. As a pre-processing step, we ℓ_2 normalize the class embeddings. Following [32, 25, 26], we use the 2048-dimensional top pooling units of a ResNet-101 pretrained on ImageNet-1K as the image representation. We do not apply any pre-processing to these features.

Evaluation. Once we train a GMN on a particular dataset $\mathcal{D}_{\text{train}}$, we synthesize n_{zsl}, $n_{\text{gzsl-u}}$ and $n_{\text{gzsl-s}}$-many samples for each unseen class to create a separate augmented dataset $\mathcal{D}^{\text{zsl}}_{\text{fake}}$, $\mathcal{D}^{\text{u}}_{\text{fake}}$, $\mathcal{D}^{\text{s}}_{\text{fake}}$ for training separate models for the ZSL, GZSL-u and GZSL-s evaluations, respectively. Additionally, we create $\mathcal{D}^{\text{a}}_{\text{fake}}$ containing n_{a} synthetic sample per unseen class, to train a single model that performs all tasks, *i.e.* classifying seen and unseen class examples. Exceptionally, only on the AWA dataset, where there is a significant imbalance among training classes, we additionally synthesize examples for the seen classes to obtain equivalent number of training

	Zero-Shot Learning			**Generalized Zero-Shot Learning**								
	CUB	**SUN**	**AWA**	**CUB**			**SUN**			**AWA**		
Method	**T-1**	**T-1**	**T-1**	**u**	**s**	**h**	**u**	**s**	**h**	**u**	**s**	**h**
Train only with real samples ($\mathcal{D}_{\mathrm{s}}$)	56.8	60.7	62.3	26.9	67.6	38.4	23.4	36.3	28.4	13.4	78.1	22.9
$\mathcal{L}^{\mathrm{S}}_{\mathrm{WGAN}} + \mathcal{L}_{\mathrm{CLS}}$	58.3	61.4	70.0	47.0	71.0	56.5	47.7	**41.2**	44.2	47.8	78.7	59.5
$\mathcal{L}^{\mathrm{S}}_{\mathrm{cWGAN}}$	60.6	62.6	72.0	55.9	71.1	62.6	53.6	41.1	46.5	55.2	79.1	65.0
$\mathcal{L}^{\mathrm{S}}_{\mathrm{WGAN}} + \mathcal{L}_{\mathrm{GM}}$	61.9	63.8	70.4	55.8	70.7	62.4	53.8	40.9	46.5	52.1	78.8	62.7
$\mathcal{L}^{\mathrm{S}}_{\mathrm{cWGAN}} + \mathcal{L}_{\mathrm{GM}}$	**64.6**	64.1	73.9	57.9	**71.2**	63.9	55.2	40.8	46.9	63.2	78.8	70.1
$\mathcal{L}^{\mathrm{S+U}}_{\mathrm{WGAN}} + \mathcal{L}_{\mathrm{GM}}$ (transductive)	**64.6**	**64.3**	**82.5**	**60.2**	70.6	**65.0**	**57.1**	40.7	**47.5**	**70.8**	**79.2**	**74.8**

Table 2: Quantitative evaluation of GMN over the strong baselines.

examples for the seen classes. These numbers are considered as hyper-parameters and therefore tuned on the validation splits. We set sample spaces in a way that they are comparable with the *ANOSPC* value of each dataset (Table 1). Then we stack each $\mathcal{D}_{\mathrm{fake}}$ together with the training set $\mathcal{D}_{\mathrm{train}}$ to form $\tilde{\mathcal{D}}$ on which we train f defined in Eq. 6. We also tune the hyper-parameters of this classifier, *i.e.* learning rate and number of training iterations, for each experimental setup separately on the validation splits as well. Once the final classifier is trained, we assign a label to a test sample by considering only the scores of unseen classes $\mathcal{Y}_{\mathrm{u}}$ for ZSL; we consider the scores of all classes $\mathcal{Y}_{\mathrm{all}}$ when determining the label for GZSL. As evaluation scores we compute normalized mean top-1 accuracies **T-1**. We compute two metrics for GZSL experiments: GZSL-u (**u**) and GZSL-s (**s**) are normalized GZSL **T-1** accuracies of unseen and seen classes, respectively . Finally we compute their harmonic means by $\mathbf{h} = \frac{2\times\mathbf{u}\times\mathbf{s}}{\mathbf{u}+\mathbf{s}}$ [32].

Implementation details. In our experiments, we realize $\mathcal{G}$ and $\mathcal{D}$ as simple MLPs that have 1 or 2 hidden layers with 1024, 2048 or 4096 units. Both networks have ReLU activation functions. We consider the dimension of noise spaces d_z as another hyper-parameter. While minimizing $\mathcal{L}_{\mathrm{GM}}$, we also update the classification model, but unlike the $\mathcal{G}$ and $\mathcal{D}$ models, we regularly re-initialize the classification model every N iterations. We tune all the hyper-parameters on the validation sets. While developing the model, we observed the followings: (i) Constraining noise means by applying tanh activation, *i.e.* $\mu(a) = \tanh(\mathcal{W}_{\mathrm{mu}}a + b_{\mathrm{mu}})$, slightly improves generalization performance. (ii) Using an identity covariance matrix, ie. $\Sigma(a) = \mathbb{I}$, performs equally well. (iii) Minimizing l_{p} loss between the gradient vectors, *i.e.* $\|g_{\mathrm{r}}(\theta) - g_{\mathrm{s}}(\theta)\|_p$ in addition to maximizing their cosine similarity leads to slightly better results. Therefore, we tune these design choices on the validation sets as well.

GMN versus strong baselines. In Table 2, we present a detailed evaluation of the GMN variants and compare them against strong baselines. In order to carefully observe the variants on each task, we train a separate model for each task following the evaluation scheme described above. The first part of the table shows baselines with no GM-loss: (i) training f with available seen class samples only, (ii) training $\mathcal{G}$ model w.r.t. unconditional discriminator based on the seen class examples ($\mathcal{L}^{\mathrm{S}}_{\mathrm{WGAN}}$) plus the loss of a pre-trained classifier ($\mathcal{L}_{\mathrm{CLS}}$), (iii) training $\mathcal{G}$ model with conditional discriminator over the seen classes ($\mathcal{L}^{\mathrm{S}}_{\mathrm{cWGAN}}$). The second part of the table shows GMN variants, where the GM-loss is used in combination with (i) an unconditional seen-class only discriminator ($\mathcal{L}^{\mathrm{S}}_{\mathrm{WGAN}}$), (ii) a conditional discriminator over the seen classes ($\mathcal{L}^{\mathrm{S}}_{\mathrm{cWGAN}}$), (iii) an unconditional discriminator over seen and unseen classes ($\mathcal{L}^{\mathrm{S+U}}_{\mathrm{WGAN}}$). In all cases (except the very first one), the resulting $\mathcal{G}$ model is used for generating n_{zsl}, $n_{\mathrm{gzsl\text{-}u}}$ and $n_{\mathrm{gzsl\text{-}s}}$-many synthetic training examples for each unseen class. Only the last experiment runs in a transductive setting. Note that we do not report any results with $\mathcal{L}^{\mathrm{S}}_{\mathrm{WGAN}}$-only or $\mathcal{L}^{\mathrm{S+U}}_{\mathrm{WGAN}}$-only training as they lead to unusable $\mathcal{G}$ due to lack any class supervision (see Fig.5).

From the results, we observe that all generative model based methods significantly improve over the simple real data only baseline. This result shows that generating unseen class examples via $\mathcal{G}$ helps training f models that are particularly better at Generalized ZSL. We also observe that using a conditional discriminator leads to better results than training with the loss function of a pre-trained classifier, as we hypothesize in Section 3.

From the results in the second part of Table 2, we observe that the proposed gradient matching loss consistently improves all ZSL and **u** scores by marginally sacrificing the **s** scores. In particular we observe that GM loss is significantly a better way for training $\mathcal{G}$ than minimizing the classification loss of a pre-trained classifier. In addition, we observe that $\mathcal{L}_{\mathrm{GM}}$ improves over the conditional discriminator based $\mathcal{G}$ training. Finally, we observe that unsupervised discriminator (over all classes) combined with the GM loss (over seen class examples only) gives a strong approach for training the conditional generative model in a transductive way.

GMN versus state of the art. Finally, we present a comparison of GMN with the recently proposed state-of-the-art non-transductive ZSL approaches on the benchmarks from

	Zero-Shot Learning			Generalized Zero-Shot Learning								
	CUB	SUN	AWA	CUB			SUN			AWA		
Method	T-1	T-1	T-1	u	s	h	u	s	h	u	s	h
Zhang et al. [46] '18	52.6	61.7	67.4	31.5	40.2	35.3	41.2	26.7	32.4	38.7	74.6	51.0
Bucher et al. [25] '17	57.8	60.4	66.3	28.8	55.7	38.0	40.5	37.2	38.8	2.3	**90.2**	4.5
Xian et al. [26] - `DEVISE` '18	60.3	60.9	66.9	52.2	42.4	46.7	38.4	25.4	30.6	35.0	62.8	45.0
Xian et al. [26] - `ALE` '18	61.5	62.1	68.2	40.2	59.3	47.9	41.3	31.1	35.5	47.6	57.2	52.0
Verma et al. [28] '18	59.6	63.4	69.5	41.5	53.3	46.7	40.9	30.5	34.9	56.3	67.8	61.5
Felix et al. [27] - `cycle-WGAN` '18	57.8	59.7	65.6	46.0	60.3	52.2	48.3	33.1	39.2	56.4	63.5	59.7
Felix et al. [27] - `cycle-CLSWGAN` '18	58.4	60.0	66.3	45.7	**61.0**	52.3	49.4	**33.6**	40.0	56.9	64.0	60.2
$\mathcal{L}^{\text{S}}_{\text{cWGAN}} + \mathcal{L}_{\text{CLS}}$ [26]	62.2	62.7	69.4	51.1	54.9	52.9	50.6	30.3	37.3	57.5	66.8	61.8
$\mathcal{L}^{\text{S}}_{\text{cWGAN}} + \mathcal{L}_{\text{GM}}$ *(Ours)*	**64.3**	**63.6**	**71.9**	**56.1**	54.3	**55.2**	**53.2**	33.0	**40.7**	**61.1**	71.3	**65.8**
$\mathcal{L}^{\text{S}}_{\text{cWGAN}} + \mathcal{L}_{\text{GM}}$ ‡ *(Ours)*	64.6	64.1	73.9	57.9	71.2	63.9	55.2	40.8	46.9	63.2	78.8	70.1

Table 3: Comparison of GMN against the baselines and the state-of-the-art on the benchmarks from [32]. The results are taken from the papers, except for [25] (using the the authors' implementation) and $\mathcal{L}^{\text{S}}_{\text{cWGAN}} + \mathcal{L}_{\text{CLS}}$ (our implementation).

[32], in Table 3. GMN, ALE and DEVISE models from [32], cycle-WGAN model from [27] and our implementation of [32] by using a bilinear compatibility function ($\mathcal{L}^{\text{S}}_{\text{cWGAN}} + \mathcal{L}_{\text{CLS}}$) employ the same generative model *i.e.* WGAN conditioned on semantic embeddings (except that GMN and $\mathcal{L}^{\text{S}}_{\text{cWGAN}} + \mathcal{L}_{\text{CLS}}$ explicitly learn a noise space conditioned on the embeddings) but optimize different auxiliary loss functions aside from the WGAN objective. Therefore, by comparing them we can see that gradient matching outperforms both minimizing the loss of a pre-trained classifier and minimizing the cycle-consistency loss in the context of zero-shot learning either when a single model is trained to perform all tasks ($\mathcal{L}^{\text{S}}_{\text{cWGAN}} + \mathcal{L}_{\text{GM}}$) or a separate model is trained to perform individual tasks ($\mathcal{L}^{\text{S}}_{\text{cWGAN}} + \mathcal{L}_{\text{GM}}$ ‡). Overall, we observe that GMN leads to significant improvements over the state of the art in terms of h-scores on all datasets.

Effect of the feature generation. Having verified the effectiveness of our framework, we now evaluate how the size of the synthetic train data affect the final ZSL and GZSL performances. For this purpose, we train two generative models on each dataset by optimizing $\mathcal{L}^{\text{S}}_{\text{WGAN}}$ and $\mathcal{L}^{\text{S+U}}_{\text{WGAN}}$, respectively. Then we create six different synthetic datasets $\tilde{\mathcal{D}}^i_u$ with each model by sampling for each class $\{10, 25, 50, 150, 200, 250\}$ features on CUB and SUN and $\{500, 600, 700, 800, 900, 1000\}$ features on AWA. In Fig. 3, we see that, as the number of features synthesized increases, **u** scores on all datasets increase rapidly, ZSL scores develop progressively on the SUN and remain almost fixed on the others. In addition, we observe a trade-off between **s** scores and the amount of synthesized features, *i.e.* **s** scores decrease dramatically as f is trained with more synthesized features. This is an expected result since (i) f is no longer biased towards seen classes, (ii) the pooled sets $\tilde{\mathcal{D}}^i$ become dominated by the synthesized sets $\tilde{\mathcal{D}}^i_u$. In fact **s** scores on the AWA support this claim *i.e.* slope of the decrease in **s** scores is much smaller, because in AWA there are 609 samples per class on average. We observe that synthesizing more features does not necessarily increase the ZSL scores. This might indicate that class specific noise spaces become less discriminative due to the WGAN updates or the generator might be learning to model feature space irrespective of the noise distributions. And finally, the gap between $\mathcal{L}^{\text{S}}_{\text{WGAN}}$ and $\mathcal{L}^{\text{S+U}}_{\text{WGAN}}$ on AWA suggests that GMN struggles in finding generalizable latent spaces when there are small set of classes and attributes.

Ablation study. We perform additional analyses to gain further insight about GMN. We begin by elaborating more on the supervision signal conveyed by $\mathcal{L}_{\text{GM}}$, when discriminator is unconditional, and the effect of utilizing samples of unseen classes on **u** scores. To do that we train four different generators over SUN dataset by optimizing $\mathcal{L}^{\text{S}}_{\text{WGAN}}$, $\mathcal{L}_{\text{GM}}$, $\mathcal{L}^{\text{S}}_{\text{WGAN}} + \mathcal{L}_{\text{GM}}$ and $\mathcal{L}^{\text{S+U}}_{\text{WGAN}} + \mathcal{L}_{\text{GM}}$, respectively. Besides, while training the generators we compute **u** scores of samples in the validation set. Not surprisingly, when a generator optimizes $\mathcal{L}^{\text{S}}_{\text{WGAN}}$ alone, it cannot distinguish class modes separately. Therefore features synthesized by this network results in a poor classifier. However, a generator optimizing only $\mathcal{L}_{\text{GM}}$ starts learning a mapping from semantic embeddings to prominent visual features. Furthermore, $\mathcal{L}^{\text{S}}_{\text{WGAN}} + \mathcal{L}_{\text{GM}}$ slightly improves the mapping by most likely modeling data manifold more effectively. Finally, as an unconditional discriminator captures statistics of samples belonging to unseen classes by means of optimizing $\mathcal{L}^{\text{S+U}}_{\text{WGAN}}$, classifier performance peaks. We share plots corresponding to each experiment in Fig. 5.

Next, we show that conditioning a discriminator on semantic embeddings and gradient matching are complementary sources of supervision for a generator network. In our experiments, we see that optimizing $\mathcal{L}^{\text{S}}_{\text{cWGAN}} + \mathcal{L}_{\text{GM}}$, compared to optimizing only $\mathcal{L}^{\text{S}}_{\text{cWGAN}}$, improves almost all perfor-

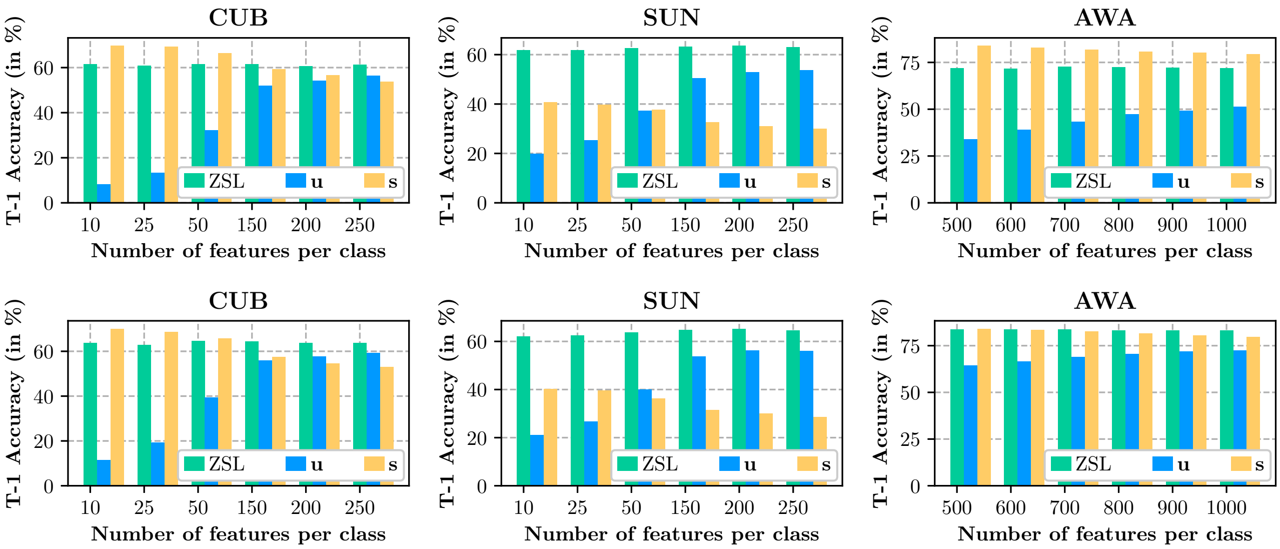

Figure 3: Analysis of the impact of the number of synthesized features on (from left to right) **T-1**, **u** and **s** scores on CUB, SUN and AWA. Top row is obtained by optimizing $\mathcal{L}^{\text{S}}_{\text{WGAN}}$, while bottom row is $\mathcal{L}^{\text{S+U}}_{\text{WGAN}}$.

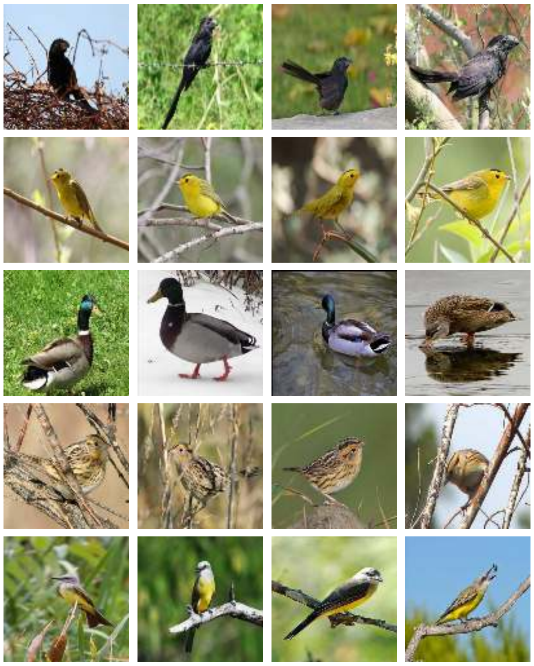

Figure 4: Examples from unseen CUB classes that are misclassified with a cWGAN-based classifier but correctly recognized when we include $\mathcal{L}_{\text{GM}}$ into generator training. From top to bottom the classes are *groove billed ani*, *wilson warbler*, *mallard*, *le conte sparrow* and *tropical kingbird*.

mance metrics significantly. We perform a rather qualitative evaluation on CUB dataset to examine the improvements that result from introducing $\mathcal{L}_{\text{GM}}$ term in a plain conditional WGAN setup. For this purpose, we inspect classes that are improved by introducing gradient matching. Significant improvements in certain classes suggest that GMN can help learning better relations between visual features and semantic embeddings. Some of examples are shown in Fig. 4.

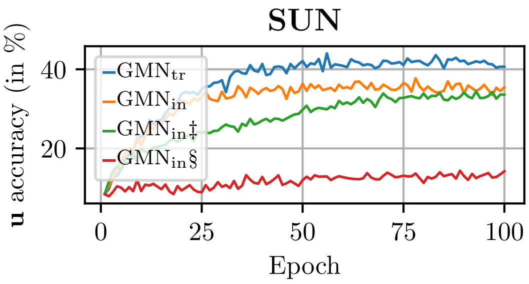

Figure 5: **u** scores of validation samples from the SUN dataset over the training iterations of $\mathcal{L}^{\text{S}}_{\text{WGAN}}$ (GMN$_{\text{in}}$§), $\mathcal{L}_{\text{GM}}$ (GMN$_{\text{in}}$‡), $\mathcal{L}^{\text{S}}_{\text{WGAN}} + \mathcal{L}_{\text{GM}}$ (GMN$_{\text{in}}$) and $\mathcal{L}^{\text{S+U}}_{\text{WGAN}} + \mathcal{L}_{\text{GM}}$ (GMN$_{\text{tr}}$). (Better viewed in color.)

5. Conclusion

In this work, we propose a novel loss function based generative model for capturing modalities of CNN features given semantic class embeddings. We show that our generative model is able to synthesize discriminative features which can be used to train a state-of-the-art classifier for generalized zero-shot classification.

Acknowledgements. This work was supported in part by the TUBITAK Grant 116E445. The numerical calculations reported in this paper were partially performed at TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA resources).

References

[1] Antonio Torralba, Rob Fergus, and William T Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2008. 1

[2] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, 2014. 1

[3] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems 29*. 2016. 1

[4] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data Augmentation Generative Adversarial Networks. *arXiv:1711.04340*, 2017. 1

[5] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to Compare: Relation Network for Few-Shot Learning. *arXiv:1711.06025*, 2017. 1

[6] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014. 1, 2

[7] Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. Zero-data learning of new tasks. In *AAAI*, 2008. 1

[8] Xiaodong Yu and Yiannis Aloimonos. Attribute-based transfer learning for object categorization with zero/one training example. In *Proc. European Conf. on Computer Vision*, 2010. 1, 2

[9] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009. 1

[10] Ali Farhadi, Ian Endres, and Derek Hoiem. Attribute-centric recognition for cross-category generalization. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010. 1

[11] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009. 1, 2, 5

[12] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 1

[13] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proc. of the Empiricial Methods in Natural Language Processing*, 2014. 1

[14] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proc. Adv. Neural Inf. Process. Syst.*, 2013. 1

[15] Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. Evaluation of output embeddings for fine-grained image classification. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015. 1, 2

[16] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for image classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016. 1

[17] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc' Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems*. 2013. 1, 2

[18] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *Advances in Neural Information Processing Systems*. 2013. 1, 2

[19] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 1, 2

[20] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *Proc. Int. Conf. Mach. Learn.*, 2015. 1, 2

[21] Yanwei Fu and Leonid Sigal. Semi-supervised vocabulary-informed learning. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 1, 2

[22] Ruizhi Qiao, Lingqiao Liu, Chunhua Shen, and Anton van den Hengel. Less is more: Zero-shot learning from online textual documents with noise suppression. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 1, 2

[23] Yanwei Fu, Timothy M Hospedales, Tao Xiang, Zhenyong Fu, and Shaogang Gong. Transductive multi-view embedding for zero-shot recognition and annotation. In *Proc. European Conf. on Computer Vision*, 2014. 1, 2

[24] Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *Proc. European Conf. on Computer Vision*, 2016. 2

[25] Maxime Bucher, Stephane Herbin, and Frederic Jurie. Generating visual representations for zero-shot classification. In *IEEE International Conference on Computer Vision Workshops*, 2017. 2, 5, 7

[26] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 2, 3, 4, 5, 7

[27] Rafael Felix, Vijay B. G. Kumar, Ian Reid, and Gustavo Carneiro. Multi-modal cycle-consistent generalized zero-shot learning. In *Proc. European Conf. on Computer Vision*, 2018. 2, 5, 7

[28] Vinay Kumar Verma, Gundeep Arora, Ashish Mishra, and Piyush Rai. Generalized zero-shot learning via synthesized examples. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 2, 5, 7

[29] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*. 2014. 2, 3

[30] 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. 2, 5

[31] Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012. 2, 5

[32] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning - the good, the bad and the ugly. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 2, 5, 6, 7

[33] Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. Zero-shot learning with semantic output codes. In *Proc. Adv. Neural Inf. Process. Syst.*, 2009. 2

[34] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg Corrado, and Jeffrey Dean. Zero-shot learning by convex combination of semantic embeddings. In *International Conference on Learning Representations*, 2014. 2

[35] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Synthesized classifiers for zero-shot learning. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 2

[36] Ziming Zhang and Venkatesh Saligrama. Zero-shot learning via semantic similarity embedding. In *Proc. IEEE Int. Conf. on Computer Vision*, 2015. 2

[37] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 2

[38] Yang Long, Li Liu, Ling Shao, Fumin Shen, Guiguang Ding, and Jungong Han. From zero-shot learning to conventional supervised classification: Unseen visual data synthesis. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 2

[39] Elyor Kodirov, Tao Xiang, Zhenyong Fu, and Shaogang Gong. Unsupervised domain adaptation for zero-shot learning. In *Proc. IEEE Int. Conf. on Computer Vision*, 2015. 2

[40] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Metric learning for large scale image classification: Generalizing to new classes at near-zero cost. In *Proc. European Conf. on Computer Vision*. 2012. 2

[41] Alina Kuznetsova, Sung Ju Hwang, Bodo Rosenhahn, and Leonid Sigal. Exploiting view-specific appearance similarities across classes for zero-shot pose prediction: A metric learning approach. In *AAAI*, 2016. 2

[42] Maxime Bucher, Stéphane Herbin, and Frédéric Jurie. Improving semantic embedding consistency by metric learning for zero-shot classiffication. In *Proc. European Conf. on Computer Vision*, 2016. 2

[43] Jie Song, Chengchao Shen, Yezhou Yang, Yang Liu, and Mingli Song. Transductive unbiased embedding for zero-shot learning. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 2

[44] 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. In *AAAI*, 2017. 2, 3

[45] Ashish Mishra, M Reddy, Anurag Mittal, and Hema A Murthy. A generative model for zero shot learning using conditional variational autoencoders. *arXiv preprint arXiv:1709.00663*, 2017. 2

[46] Haofeng Zhang, Yang Long, Li Liu, and Ling Shao. Adversarial unseen visual feature synthesis for zero-shot learning. *Neurocomputing*, 2018. 2, 5, 7

[47] Y. Long, L. Liu, F. Shen, L. Shao, and X. Li. Zero-Shot Learning Using Synthesised Unseen Visual Data with Diffusion Regularisation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2018. 2

[48] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*. 2017. 2, 3

[49] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proc. Int. Conf. Mach. Learn.*, 2017. 3

[50] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Proc. Int. Conf. Learn. Represent.*, 2014. 3

[51] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proc. Int. Conf. Mach. Learn.*, 2014. 3

[52] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier GANs. In *Proc. Int. Conf. Mach. Learn.*, 2017. 4

[53] Takeru Miyato and Masanori Koyama. cGANs with projection discriminator. In *Proc. Int. Conf. Learn. Represent.*, 2018. 4

[54] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 5