Towards Zero-Shot Learning with Fewer Seen Class Examples

Vinay Kumar Verma[1*] Ashish Mishra[2*] Anubha Pandey[2$]
Hema A. Murthy[2] Piyush Rai[1]
[1]Department of CSE, IIT Kanpur; [2]Department of CSE, IIT Madras,
{vkverma,piyush}@cse.iitk.ac.in; {mishra,hema}@cse.iitm.ac.in; anubhap93@gmail.com

Abstract

We present a meta-learning based generative model for zero-shot learning (ZSL) towards a challenging setting when the number of training examples from each seen *class is very few. This setup contrasts with the conventional ZSL approaches, where training typically assumes the availability of a sufficiently large number of training examples from each of the seen classes. The proposed approach leverages meta-learning to train a deep generative model that integrates variational autoencoder and generative adversarial networks. We propose a novel task distribution where meta-train and meta-validation classes are disjoint to simulate the ZSL behaviour in training. Once trained, the model can generate synthetic examples from seen and unseen classes. Synthesize samples can then be used to train the ZSL framework in a supervised manner. The meta-learner enables our model to generates high-fidelity samples using only a small number of training examples from seen classes. We conduct extensive experiments and ablation studies on four benchmark datasets of ZSL and observe that the proposed model outperforms state-of-the-art approaches by a significant margin when the number of examples per seen class is very small.*

1. Introduction

The traditional machine learning models for supervised classification assume the availability of labeled training examples from all the classes. This requirement is unrealistic and is rarely true in many real-world classification problems that consist of an ever-growing set of classes. Zero-Shot Learning (ZSL) [10, 39] is a learning paradigm aimed at addressing this issue and learns to predict labels of inputs from the previously unseen classes. In particular, ZSL assumes that, during training, we have access to labeled examples from a set of *seen* classes, and the test examples come from *unseen* classes, i.e., classes that were not present during training. The ZSL methods typically accomplish learning using the class attributes/descriptions of the seen and unseen classes that help bridge the two.

Recent work on ZSL has shown the effectiveness of deep learning-based approaches [39, 58, 20, 5, 44, 52, 23, 4, 59, 22, 6, 50]. However, these methods usually assume that a large number of labeled example from each of the seen classes are available to train the model. In practice, it may often be challenging and time-consuming to collect a large number of labeled examples from the seen classes; sometimes, very few (say 5-10) labeled examples per seen classes may be available. Recently, meta-learning based frameworks [9, 48, 33] have addressed the issue of labeled data scarcity and have shown promising results for supervised few-shot learning problems[31, 34, 42, 17, 21]. The basic idea in meta-learning is to train a model on various learning tasks, such that it can solve a new learning task using only a small number of training samples from each task. However, the meta-learning framework cannot handle the scenario if no labeled data is available for the unseen classes (new tasks). In this work, we propose incorporating the ZSL setup in meta-learning to handle this issue.

The recent works on ZSL, generative models have gained a significant attention [44, 52, 5, 24, 45] and have shown promising results for ZSL as well as for the more challenging setting of ZSL called *generalized* zero-shot learning (GZSL). In GZSL, the test inputs may be from seen as well as unseen classes. The success of generative models for ZSL leverages over recent advances in deep generative models, such as VAE [18] and GAN [11, 2]. These generative models [44, 52, 5, 24, 45] can generate the *synthetic* examples from unseen and seen classes (given their class attributes), using which we can train a classifier, which essentially turns the ZSL problem into a supervised learning problem. The generative approach is appealing but suffers from several shortcomings: (i) These models do not mimic the ZSL behavior in training; therefore, there may be a large gap between the original and generated samples' quality. (ii) These methods require a significant amount of labeled data, and (iii) GAN based models generate high-quality samples but suffer from mode collapse. In contrast, VAE based models generate diverse samples, but there is a large

*Equal contribution. $ Currently author is affiliated with MasterCard

gap between generated samples and the actual samples.

We develop a meta-learning-based ZSL framework that integrates a conditional VAE (CVAE) and a conditional GAN (CGAN) architecture to address the issues mentioned above. We divide the dataset into task/episode where, unlike standard meta-learning, each task has *disjoint* support-set and query-set classes. The disjoint classes help to mimic the ZSL behaviour during training. The joint architecture helps to overcome the mode collapse problem and generates diverse samples for the seen and unseen classes. The augmentation of the joint architecture of VAE-GAN, with the meta-learning model, helps to train the model even when very few (say 5 or 10) samples per seen class are available. Once trained, the model can synthesize unseen class samples. These synthesized samples can be used to train any supervised classifier, essentially turning ZSL into a supervised learning problem. Our generation based framework is also ideal for the *generalized* ZSL problem [23, 44, 3, 15, 45, 54] where the test inputs can be from seen as well as unseen classes.

Developing a meta-learning approach for ZSL is challenging. In the supervised meta-learning setup [9, 48, 33], we have meta-train, meta-validation, and meta-test. The meta-train and meta-validation share the same classes, while meta-test classes are disjoint. Also, the meta-test contains a few labeled samples for each unseen class. In the ZSL setup for the meta-test, we do not have any labeled samples but have access only to the class attribute vector. Contrary to the standard meta-learning approach, the meta-train and meta-validation classes are disjoint in the proposed ZSL setting. The disjoint class simulates the ZSL setup during training. For the task distribution of the data, we follow the N-way, K-shot setup proposed by [48] and trained our model under Model-Agnostic Meta-Learning (MAML) [9] framework that adapted to the ZSL setting. The main contributions of the proposed approach are summarized as follows:

- We propose a meta-learning based framework for ZSL and generalized ZSL problem. We train the joint VAE-GAN model with a meta-learner's help to generate diverse, high-quality samples that yield superior ZSL prediction accuracy.

- We propose a novel task distribution different from the traditional meta-learning model [9, 48] that mimics the ZSL setup in training and helps generate robust unseen class samples.

- Our framework can learn even when only a few labeled examples (say 5 or 10) are available per seen class. On the standard datasets, our approach shows state-of-the-art results against the strong generative baselines for the ZSL and GZSL.

2. Related Work

Some of the earliest approaches for ZSL are based on learning a mapping from visual space to semantic space. During the test time, they use the learned mapping to predict the class-attributes for unseen examples and then do a nearest neighbor search to predict the class label [20, 27, 39]. Similarly, some methods also use transfer learning between seen classes and unseen classes. They represent the parameters of unseen classes as a similarity weighted combination of the parameters of seen classes [35, 58]. All these approaches rely on the availability of plenty of labeled data from the seen classes and do not perform well in the GZSL setting.

Another popular approach to ZSL focuses on learning the linear/bilinear compatibility function between the visual and semantic spaces. Methods, such as ALE [1], DEVISE [10], ESZSL [35],[43], learn a relationship between the inputs and class-attributes via a linear compatibility function, whereas SAE [19] adds an autoencoder loss to the projection that encourages re-constructability from the attribute space to the visual space. Some of the approaches [41, 55, 30] assume that all test inputs of unseen classes are present during training and use these *unlabeled* examples also in training. This setting is referred to as *transductive* setting because of the extra information model leads to improve performance. The transductive assumption is often not realistic since the test inputs are usually not available during training. The attention-based approach is also explored in [14, 60], and these approaches work well for the fine-grain datasets. Paper [12, 23] are calibration based approach and [12] use metric based meta-learning for the feature calibration.

The Generalized ZSL (GZSL) problem is a more realistic and challenging problem as compared to standard ZSL. Unlike standard ZSL, in GZSL, the seen (training) and unseen (test) classes are not disjoint, which makes ZSL harder since the classifier is biased towards classifying each input (seen/unseen class) as belonging to the seen class. Most of the previous approaches perform well in standard ZSL but fail to handle biases towards seen classes [1, 35, 19, 10]. Recently, generative models have shown promising results for both ZSL and GZSL setting. These approaches synthesize examples from both seen and unseen classes to train a supervised classifier, which somewhat mitigates the bias towards seen classes [47, 5, 24, 8, 52, 36, 13, 30, 37]. Most of the recent generative models for ZSL are based on VAE [40] and GAN [11]. Among the generative models, [24, 44, 37, 50, 46, 56] are based on VAE architectures, while [57, 26, 52, 8, 5, 13, 45, 29, 28, 25, 16] use adversarial learning for sample generation based on the class-attribute. The current generative approaches are unable to mimic ZSL behaviour in training. This limits the model's performance; also, they require significant labeled data from seen classes.

In contrast, our proposed model Meta-VGAN leverage on the meta-learning framework can easily learn using very few labeled examples (as few as 5-10) from the seen classes. Also, disjoint task distribution helps to mimic ZSL behaviour in training. Our proposed model mainly focuses on ZSL/GZSL when very few labeled examples from seen classes are available. In contrast, we also conduct experiments using all examples from seen classes and observe that the proposed approach outperforms all recent state-of-the-art methods.

3. Proposed Model

3.1. Problem Setup

In ZSL, we assume that classes are divided into a disjoint set of seen (train) and unseen (test) classes. Only seen class examples are present during training, while the test set contains examples form unseen classes. Let S and U denote the number of seen/train and unseen/test classes, respectively. In ZSL, we also assume having access to class-attribute vectors/descriptions $\mathcal{A} = \{\mathbf{a_c}\}_{c=1}^{S+U}$, $\mathbf{a_c} \in \mathbb{R}^d$ for each of the seen and unseen classes. Our ZSL approach is based on synthesizing labeled features from each (unseen/seen) class using the trained generative model given the class attribute. Later, synthesized label features can be used to train a classifier; essentially, this turns the ZSL problem to a standard supervised learning problem.

Task-Distribution: We use the MAML [9] as the base meta-learner. Note that unlike few-shot learning, in ZSL, no labeled data are available for the unseen classes. Therefore we also need to modify the episode-wise training scheme used in traditional MAML. We have train and test samples for ZSL where train and test classes are disjoint, also each class are associated with an attribute vector. The training data further divide into train-set and validation-set. Meta-learning further divides the train-set into a set of tasks following the N-way and K-shot setting, where each task contains N classes and K samples for each class. In particular, let $P(\mathcal{T})$ be the distribution of tasks over the train-set i.e. $P(\mathcal{T}) = \{\tau_1, \tau_2, \ldots, \tau_n\}$, τ_i is a task and $\tau_i = \{\tau_i^v, \tau_i^{tr}\}$. The set τ_i^v and τ_i^{tr} are in the N-way K-shot setting. However, unlike the traditional meta-learning setup, the classes of τ_i^v and τ_i^{tr} are *disjoint*. The disjoint class split simulate the ZSL setting during training and helps to generate high-quality unseen class samples.

3.2. Motivation for Zero-Shot Meta-Learning

The standard meta-learning [9, 48, 38, 33] can quickly adapt to novel tasks using only a few samples. As demonstrated in prior work [48, 38], even without any labeled data form the novel classes, the meta-learning models can learn robust discriminative features. These works have also shown that sometimes zero-gradient step (without fine-tuning) model performs better than the fine-tuned architecture (refer to Table 1 in [48]). Another recent work [32] demonstrates that the meta-learning framework is more about feature reuse instead of quick adaption. They have shown that a trained meta-learning model can provide a good enough feature for the novel classes that secure high classification accuracy without any fine-tuning. This motivates to a train of a generative model (VAE, GAN) in the meta-learning framework to synthesize the novel class samples. Our approach is similar in spirit to [32]. The main difference is that (i) the proposed approach learns a generative model instead of discriminative, and (ii) Our problem setting is ZSL as opposed to a few-shot classification. In particular, the proposed approach's objective is to solve ZSL by generating novel class samples, while the goal in [32] is to learn discriminative features that are suitable for (few-shot) classification.

3.3. Meta Learning-based Generative Model

The proposed generative model (Meta-VGAN), shown in Fig. 1, is a combination of conditional VAE (CVAE) and conditional GAN (CGAN). The model has shared parameters between the decoder and the GAN generator. The decoder's reconstructed samples are also fed to the discriminator; this increases the decoder's generation robustness. The decoder implicitly also works as a classifier. The Meta-VGAN consists of four modules - Encoder (E), decoder (De), Generator (G), and Discriminator (Ds). Each of these modules is augmented with a meta-learner. The E and De modules are from CVAE, whereas G and Ds modules are formed CGAN. Here De and G share common network parameters over different inputs. Unlike other generative models for ZSL [52, 8], we do not need a classifier over the G module, since De ensure the separability of the different classes generated by G. Let the parameters of the module E and Ds be denoted by θ_e and θ_d and the shared parameter of De and G be denoted by θ_g. In Fig.-1, we have shown the module-wise overall structure of the proposed model. The objective function of CVAE is defined as:

$$\begin{aligned}\mathcal{L}^V(\theta_e, \theta_g) = &- KL(q_{\theta_e}(\mathbf{z}|\mathbf{x}, \mathbf{a_c})||p(\mathbf{z}|\mathbf{a_c})) \\ &+ \mathrm{E}_{\mathbf{z}\sim q_{\theta_e}(\mathbf{z}|\mathbf{x},\mathbf{a_c})}[\log p_{\theta_g}(\mathbf{x}|\mathbf{z}, \mathbf{a_c})]\end{aligned} \tag{1}$$

Here $\mathbf{a_c} \in \mathbb{R}^k$ is the attribute vector of the class for the input $\mathbf{x} \in \tau_i^{tr}$, and $\tau_i^{tr} \in \tau_i$. The conditional distribution $q_{\theta_e}(\mathbf{z}|\mathbf{x}, \mathbf{a_c})$ is parametrized by the encoder $E(\mathbf{x}, \mathbf{a_c}, \theta_e)$ output, the prior $p(\mathbf{z}|\mathbf{a_c})$ is assumed to be $N(0, \mathbf{I})$, and $p_{\theta_g}(\mathbf{x}|\mathbf{z}, \mathbf{a_c})$ is parametrized by the decoder $De(\mathbf{z}, \mathbf{a_c}, \theta_g)$ output. Similarly, for CGAN, the discriminator objective $\mathcal{L}^D$ and generator objective $\mathcal{L}^G$ are

$$\begin{aligned}\mathcal{L}^D(\theta_g, \theta_d) =&\mathbb{E}[Ds(\mathbf{x}, \mathbf{a_c}|\theta_d)] - \mathbb{E}_{\mathbf{z}\sim N(0,1)}[Ds(G(\mathbf{z}, \mathbf{a_c}|\theta_g), \mathbf{a_c}|\theta_d)] \\ &- \mathbb{E}_{\mathbf{z}\sim N(0,1)}[Ds(De(\mathbf{z}, \mathbf{a_c}, \theta_g)), \mathbf{a_c}|\theta_d)]\end{aligned} \tag{2}$$

$$\mathcal{L}^G(\theta_g, \theta_d) = \mathbb{E}_{\mathbf{z}\sim N(0,\mathbf{I})}[Ds(G(\mathbf{z}, \mathbf{a_c}|\theta_g), \mathbf{a_c}|\theta_d)] \tag{3}$$

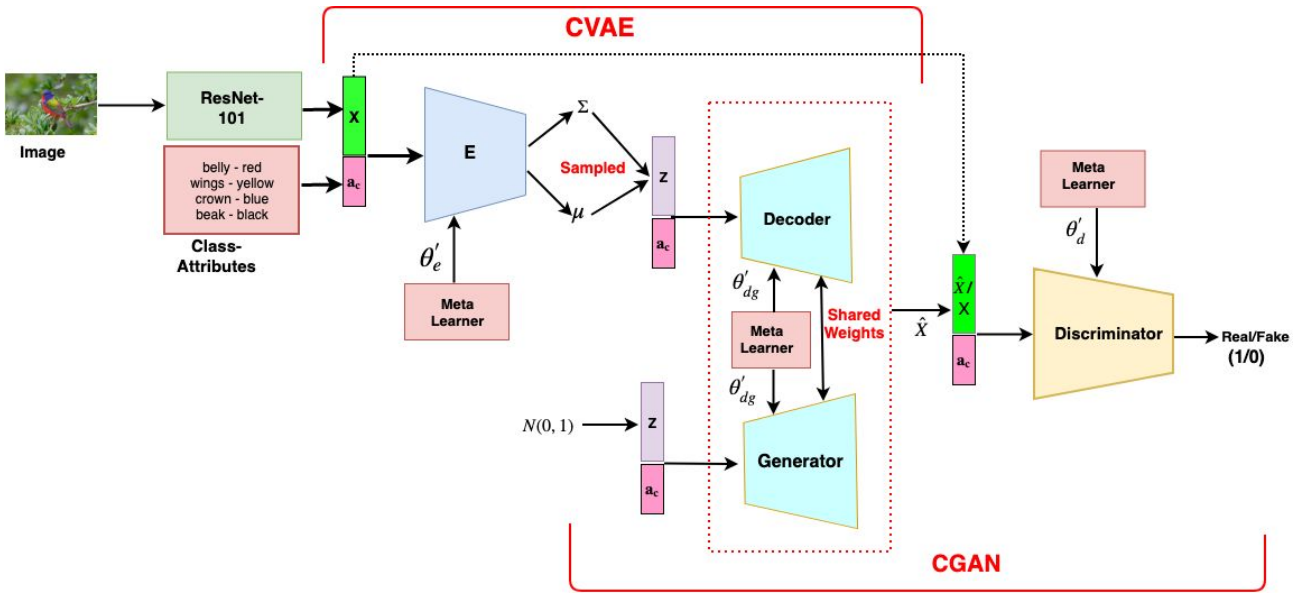


Figure 1: The overall architecture for our Meta-VGAN Zero Shot Learning model

Next, we adapt the above objective to the MAML framework over the proposed task distribution (Sec. 3.1). Our approach samples a task τ_i from the task distribution $P(\mathcal{T})$ i.e. $\tau_i \sim P(\mathcal{T})$ where $\tau_i = \{\tau_i^v, \tau_i^{tr}\}$. The τ_i^{tr} is used to train the inner loop of the joint objective and τ_i^v is used to update the meta-objective. The joint objective of the CVAE and generator G for the task τ_i is defined as:

$$l_{\tau_i}^{VG} = \min_{\theta_{eg}} \left(-\mathcal{L}^V(\theta_e, \theta_g) + \mathcal{L}^G(\theta_g, \theta_d)\right) \tag{4}$$

where $\theta_{eg} = [\theta_e, \theta_g]$ collectively denotes all the parameters of E and G/De. The discriminator Ds considers the samples generated from the G and De as fake samples and samples obtained from τ_i as real samples. Its objective for each task τ_i is given as:

$$l_{\tau_i}^{D} = \max_{\theta_d} \mathcal{L}^D(\theta_g, \theta_d) \tag{5}$$

In Eq. 4-5 τ_i is τ_i^{tr} for the inner loop and τ_i^v for the global update. A brief description is provided below. The meta-learner (inner loop) update are performed on $\tau^{tr} \in \tau_i$. Empirically, for the ZSL setting, which is considerably harder than standard meta-learning, we observe that updates over individual tasks are unstable during training (because of GAN). To solve this problem, unlike the standard meta-learning model where updates are performed on each task, we update the meta-learner over a batch of tasks, i.e., the loss is averaged out over the set of tasks. Therefore, our inner loop objective is given by:

$$\min_{\theta_{eg}} \sum_{\tau_i^{tr} \in \tau_i \sim P(\mathcal{T})} l_{\tau_i^{tr}}^{VG} \quad \text{and} \quad \max_{\theta_d} \sum_{\tau_i^{tr} \in \tau_i \sim P(\mathcal{T})} l_{\tau_i^{tr}}^{D} \tag{6}$$

The inner loop update for Eq. 4, i.e. joint objective of CVAE and G is performed as:

$$\theta'_{eg} \leftarrow \theta_{eg} - \eta_1 \nabla_{\theta_{eg}} \sum_{\tau_i^{tr} \in \tau_i \sim P(\mathcal{T})} l_{\tau_i^{tr}}^{VG}(\theta_{eg}) \tag{7}$$

Similarly the inner loop update for Eq. 5 i.e. Ds's objective is performed as:

$$\theta'_{d} \leftarrow \theta_{d} + \eta_2 \nabla_{\theta_{d}} \sum_{\tau_i^{tr} \in \tau_i \sim P(\mathcal{T})} l_{\tau_i^{tr}}^{D}(\theta_{d}) \tag{8}$$

The optimal parameters obtained by meta-learner (θ'_{eg} and θ'_d) are further applied on the meta-validation set τ_i^v. Note that, since classes of τ_i^v and τ_i^{tr} are disjoint, in the outer loop the optimal parameters θ'_{eg} and θ'_d are applied to *novel* classes. With these parameters, if the model can generate novel class samples that fool Ds, it indicates that the model has the ability to generate novel class samples. Otherwise, the outer loop is updated, and θ'_{eg} and θ'_d are considered the initializer for the model on the outer loop. The outer loop objective for the Eq. 4 and Eq. 5 is given by:

$$\min_{\theta_{eg}} \sum_{\tau_i^{v} \in \tau_i \sim P(\mathcal{T})} l_{\tau_i^{v}}^{VG}(\theta'_{eg}) \quad \& \quad \max_{\theta_d} \sum_{\tau_i^{v} \in \tau_i \sim P(\mathcal{T})} l_{\tau_i^{v}}^{D}(\theta'_{d}) \tag{9}$$

The global update (outer loop) for the Eq. 9 is:

$$\theta_{eg} \leftarrow \theta_{eg} - \eta_1 \nabla_{\theta_{eg}} \sum_{\tau_i^{v} \in \tau_i \sim P(\mathcal{T})} l_{\tau_i^{v}}^{VG}(\theta'_{eg}) \tag{10}$$

$$\theta_{d} \leftarrow \theta_{d} + \eta_2 \nabla_{\theta_{d}} \sum_{\tau_i^{v} \in \tau_i \sim P(\mathcal{T})} l_{\tau_i^{v}}^{D}(\theta'_{d}) \tag{11}$$

The algorithm 1 represents the training procedure of the proposed Meta-VGAN model.

3.4. Zero-Shot Classification using Synthesized Examples

Once the Meta-VGAN model is trained, we synthesize the examples (both unseen or seen classes) using their respective class attribute vectors passed to the generator/decoder module ($G_{\theta_{dg}}$). The generation of unseen class examples is done as $\hat{\mathbf{x}} = G_{\theta_{dg}}(\mathbf{z}, \mathbf{a_c})$. Here $\mathbf{z}$ is sampled from a unit Gaussian i.e. $\mathbf{z} \sim N(0, \mathbf{I})$. The sampled $\mathbf{z}$ is concatenated with the class-attributes $\mathbf{a_c}$ and passed to the decoder as input, and it generates $\hat{\mathbf{x}}$, i.e., feature vectors of input from class $\mathbf{c}$. These synthesized unseen class examples can then be used as labeled examples to train any supervised classifier (e.g., SVM/Softmax). In the GZSL setting, we synthesize *both* the seen and the unseen class examples using the class attributes and train an SVM or softmax classifier on all these training examples. The Experiments section contains further details of our overall procedure.

Algorithm 1 Meta-VGAN for ZSL

Require: $p(\mathcal{T})$:distribution over tasks
Require: η_1, η_2: step-size hyperparameters
1: Randomly initialize $\theta_e, \theta_g, \theta_d$
2: **while** not done **do**
3: Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$; where, $\mathcal{T}_i = \{\mathcal{T}_i^{tr}, \mathcal{T}_i^{v}\}$ such that $\mathcal{T}_i^{tr} \cap \mathcal{T}_i^{v} = \phi$
4: **for all** $\mathcal{T}_i$ **do**
5: Evaluate $\nabla_{\theta_{eg}} l^{VG}_{\mathcal{T}_i^{tr}}(\theta_{eg})$
6: Evaluate $\nabla_{\theta_d} l^{D}_{\mathcal{T}_i^{tr}}(\theta_d)$
7: Compute adapted parameters: $\theta^{'}_{eg} = \theta_{ed} - \eta_1 \nabla_{\theta_{eg}} l^{VG}_{\mathcal{T}_i^{tr}}(\theta_{eg})$
8: Compute adapted parameters: $\theta^{'}_{d} = \theta_{d} + \eta_2 \nabla_{\theta_d} l^{D}_{\mathcal{T}_i^{tr}}(\theta_d)$
9: Update $\theta_{eg} \leftarrow \theta_{eg} - \eta_1 \nabla_{\theta_{eg}} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} l^{VG}_{\mathcal{T}_i^{v}}(\theta^{'}_{eg})$
10: Update $\theta_{d} \leftarrow \theta_{d} + \eta_2 \nabla_{\theta_d} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} l^{D}_{\mathcal{T}_i^{v}}(\theta^{'}_{d})$

Dataset	Attribute/Dim	#Image	Seen/Unseen Class
AwA2[51]	A/85	37322	40/10
CUB[49]	CR/1024	11788	150/50
SUN[53]	A/102	14340	645/72
aPY[7]	A/64	15339	20/12

Table 1: The benchmark datasets used in our experiments, and their statistics.

4. Experiments

We conduct experiments on four widely used benchmark datasets and compare our approach with several state-of-the-art methods. Our datasets consist of Animals with Attributes (AwA) [51], aPascal and aYahoo (aPY) [7] , Caltech-UCSD Birds-200-2011 (CUB- 200) [49] and SUN Attribute (SUN-A) [53]. The Table 1 shows description about the datsets.The details description of the datasets are provided in the supplementary material. We report our results on ZSL evaluation metrics proposed by [51]. In particular, for GZSL, the harmonic mean of the seen and unseen class accuracies is reported, while for ZSL, the mean of the per-class accuracy is reported. These evaluation metrics helps to evaluate the model in an unbiased way. ResNet-101 features are used for all the datasets. All the baselines models also use the same features and evaluation metrics. Due to space limitations, further details of the implementation and experimental setting are provided in the supplementary material.

4.1. Comparison with ZSL baselines

To prove our proposed model's efficacy, we create a baseline for a few examples per seen class using various recent state-of-the-art models. We follow the same experimental settings and the same number of samples for all the baseline as we used in the proposed model.

- **CVAE-ZSL [24]**: CVAE-ZSL is a conditional variational autoencoder based generative model converts a ZSL problem into a typically supervised classification problem by using synthesized labeled inputs for the unseen classes. This approach generates the samples conditioned on the unseen class attributes, using a CVAE. The generated samples are used to train a classifier for the unseen classes. For the GZSL setting, both seen and unseen class samples are generated, and the classifier is trained on the union set.

- **GF-ZSL** [44] GF-ZSL, a state-of-the-art method, is an extended version of the CVAE-ZSL model, where it uses a feedback mechanism in which a discriminator (a multivariate regressor) learns to map the generated samples to the corresponding class attribute vectors, leading to an improved generator.

- **f-CLSWGAN [52]** f-CLSWGAN uses the Wasserstein-GAN as a generative model with an additional classifier associated with the generator. The classifier increases the class separation between two classes, while the generator's goal is to generate samples that follow the original data distribution closely.

- **cycle-UWGAN [8]** This approach uses cycle consistency loss as a regularizer with Wasserstein-GAN, which helps to reconstruct back semantic features from generated visual features. Using cycle consistency loss, it synthesizes more representative visual features for both seen and unseen classes.

- **ZSML [45]** Recently ZSML [45] train the GAN in the meta-learning framework, that helps to generate

the robust samples for the novel classes. The proposed approach shows a significant improvement over the ZSML model.

Method	N	SUN	CUB	AwA2	APY
CVAE-ZSL [24]	5	49.7	52.8	41.5	21.6
	10	51.2	53.5	45.6	21.8
GF-ZSL [44]	5	53.2	56.5	57.4	32.6
	10	55.3	56.9	59.2	35.1
cycle-UWGAN [8]	5	45.6	56.8	61.9	36.3
	10	45.8	57.2	62.0	38.1
f-CLSWGAN [52]	5	32.6	40.9	59.8	41.4
	10	34.5	41.5	62.3	41.9
Ours(Meta-VGAN)	5	**59.1**	**66.9**	**67.3**	**48.6**
	10	**60.3**	**68.8**	**68.4**	**49.2**

Table 2: Zero-Shot learning (ZSL) results only using five and ten example per seen classes to train the model

4.2. Results

We train our model and all the baselines using five and ten examples per seen class. For this experiment, we compare our model with CVAE-ZSL, GF-ZSL, f-CLSWGAN, and cycle-UWGAN. We consider both the ZSL (test examples only from unseen classes) as well as generalized ZSL (test examples from seen as well as unseen classes) settings. To prove our proposed Meta-VGAN model's efficacy, we also perform the experiments using all the examples from seen classes and compare against recent methods (refer Table-4).

Zero-Shot Learning (ZSL)

For the ZSL setting, we compare our Meta-VGAN with the state-of-the-art methods on all the datasets in Table 2. The proposed approach shows a significant improvement over the baseline methods for all four datasets. We use a small training set (5 and 10 samples per seen class) to train the model in all of these experiments. We observe that our Meta-VGAN model can train the model well using only a few examples per seen class and significantly better results than the baselines. The proposed approach achieves the substantial absolute performance gains by more than 11.1%, 5.4%, 7.2%, and 6.4%, in comparison with the baseline methods, on CUB, AWA2, aPY, and SUN datasets, respectively for five examples per seen class.

Generalized Zero-Shot Learning (GZSL)

In the GZSL settings, the test input is classified into the joint class space of the seen and unseen classes, i.e., $Y = Y^S \cup Y^U$. The GZSL setting is more practical as it removes the assumption that test input only comes from unseen classes. For the evaluation metric, we compute the harmonic mean [51] of seen and unseen class accuracy's: $H = 2 * S * U/(S + U)$, where S and U denote the accuracy of seen classes and unseen classes respectively. We evaluate our method on three standard datasets and show the performance comparison with the baseline approach in Table 3.

To have a fair comparison with the previous state-of-the-art generative methods, we keep the number of synthetic samples per class the same. For the baselines, we follow the same setup as in the original papers and perform experiments in our novel setup when only a few samples per seen class are available. We perform the experiments in two settings. In the first setting, we assume that each seen class has only five samples for training, while in the other setting, we assume that ten samples are available. In both cases, our Meta-VGAN method shows the improvements by notable margins on all the datasets compared to all baseline methods. Our model achieves the substantial absolute performance gains in the harmonic mean by 10.8%, 11.9%, and 3.4% in comparison with the baseline method on CUB, AWA2, and aPY datasets, respectively.

5. Ablation Study

To disentangle the role of each component of the proposed framework, we conduct a detailed ablation analysis. We are primarily interested in the following: (i) The performance with and without meta-learner. (ii) The performance in traditional ZSL when the number of examples per seen class is *not* very small. (iii) Investigating why meta-learner helps in improved sample generation and which of the components of our model benefit more from meta-learning and (iv) The effect of disjoint task distribution vs standard task distribution.

5.1. With and without meta-learning

The proposed model trains the generative model in the meta-learning framework. To illustrate the contribution of the meta-learner module, we perform an experiment when no meta-learning component is present. We trained the model with and without meta-learner, for CUB and AWA2 datasets, for standard ZSL and GZSL settings. We observe that a meta-learner's role in our proposed model is crucial, which helps to train the model very efficiently using a few examples per seen class. The meta-learner component boosts the model's absolute performance by 7.8% and 6.4% in standard ZSL, while by 9.9% 6.2% in GZSL for CUB and AWA2 datasets, respectively. Please refer to supplementary material for more detail.

5.2. What if we DO have plenty of training examples from seen classes?

Our meta-learning-based model is primarily designed for the setting when the number of training examples per seen class is very small (e.g., 5-10). We also investigate whether meta-learning helps when the number of examples per seen class is *not* small (i.e., the conventional ZSL setting). In this experiment, we consider the complete dataset for our

Method	N	AwA2			CUB			aPY		
		U	S	H	U	S	H	U	S	H
CVAE-ZSL [24]	5	16.5	28.7	21.0	50.1	28.1	36.0	19.9	65.4	30.5
	10	18.1	29.2	22.3	50.5	28.7	36.5	20.5	65.8	31.2
cycle-UWGAN [8]	5	40.4	43.3	41.8	48.2	33.3	39.4	18.6	64.2	28.8
	10	45.5	50.9	48.0	48.3	35.2	40.7	19.6	66.3	30.2
f-CLSWGAN [52]	5	37.8	44.2	40.7	30.4	28.5	29.4	17.2	40.6	24.5
	10	40.5	55.9	46.9	34.7	38.9	36.6	21.2	32.2	25.6
GF-ZSL [44]	5	38.2	44.3	41.0	29.4	33.0	31.0	20.2	66.3	30.9
	10	41.4	45.9	43.5	35.6	43.5	39.1	20.6	67.8	31.5
ZSML [45]	5	38.4	61.3	47.3	32.9	38.2	35.3	–	–	–
	10	47.8	59.6	53.1	42.7	45.1	43.9	–	–	–
Ours (Meta-VGAN)	5	44.5	67.5	**53.7**	51.5	47.8	**50.2**	24.1	58.9	**34.3**
	10	46.2	73.1	**56.6**	52.5	49.2	**52.1**	24.5	59.1	**35.3**

Table 3: GZSL results when only five and ten examples per seen class are used to train the model. We randomly selected 5/10 samples per class, for examples in the CUB dataset for 150 training classes we have new dataset size 150*(5/10), rest samples are not used.

experiment; this is essentially a traditional ZSL setting.

We conduct experiments on both standard ZSL and GZSL setting for CUB and AWA2 datasets, and the results are shown in Table 4. Using all the examples from seen classes to train the model, we observe that our model improves the result with a significant margin compared to all baseline approaches in both the settings. For the ZSL setting, our model achieves 9.4% and 2.1% improvement as compared to the state-of-the-art result on CUB and AWA2 datasets, respectively. Similarly, for the GZSL setting, the model achieves consistent performance gain harmonic mean (a more meaningful metric) on CUB and AWA2 datasets. We observe that all existing baseline methods show a significant difference in performance between the two regimes, i.e., using all samples and using only a few samples. In contrast, our proposed model shows competitive results in both cases. It shows that the existing baseline approaches are not suitable when the number of seen class examples is very small.

5.3. Why meta-learning helps and which model components benefit by it the most?

The meta-learning framework is capable of learning using only a few samples from each seen class. The episode-wise training helps the meta-learning framework to adapt to the new task quickly using only a few samples. In the zero-shot setup, we do not have the samples of the unseen classes; therefore, quick adaption is not possible in this case. In our proposed approach, we trained the model such that adaption is based solely on the class attribute vectors. Later, the unseen class attribute vectors can be used for unseen class sample generation. In the case of adversarial training, the generation quality depends on the quality of discriminator, generator, and the feedback provided by discriminator to the generator. The inner loop of the meta-learner provides reliable guidance (initialization) to the outer loop. As we know, adversarial training is sensitive to initialization, in our case, because of the better initialization model learns the robust parameter. The guided discriminator network is capable of learning the optimal parameters with the help of a very few examples. Therefore discriminator provides strong feedback to the generator. The guided (better initialized) generator uses this strong feedback, and because of its improved power, the generator can counter the discriminator. The alternating optimization with the guided generator and discriminator improves the overall learning capability of the adversarial network. Similarly, the meta-learning framework also helps the VAE architecture using the inner loop guidance to initialize the parameter. Therefore the joint framework in the proposed approach improves the generation quality even though only very few samples are available per seen class.

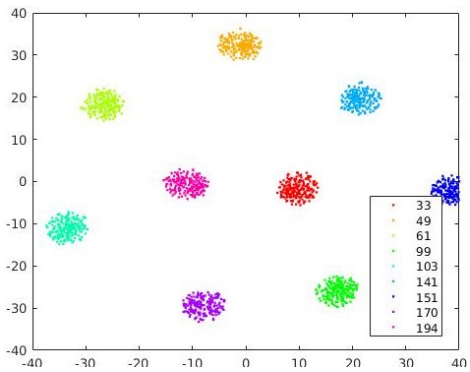
We also experiment with the discriminator without meta-learning. On CUB, this reduces the ZSL accuracy from 70.4% to 65.6%. The performance drop occurs because, without meta-learning, the discriminator is sub-optimal and is unable to provide robust feedback to the generator. Therefore the generator is unable to learn robustly. Also, we observe that removal of meta-learner from the generator (but not the discriminator), over CUB dataset, the performance drops from 70.4% to 68.8%. Through this analysis, we observe that the meta-learning based discriminator is more critical than the meta-learning based generator.

5.4. Disjoint task distribution vs. standard task distribution

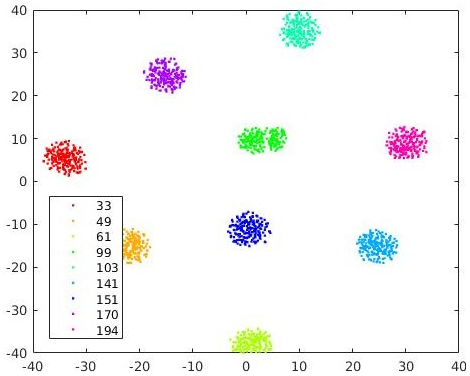
The empirical evaluation shows that the proposed disjoint task distribution helps to generate robust samples for the seen/unseen classes. We have disjoint classes between τ_i^{tr} and τ_i^{v}. Therefore, if the inner loop parameter is optimal only for the τ_i^{tr}, it produces a high loss for the τ_i^{v}. Consequently, the network tries to learn an optimal param-

Method	Standard Setting		Generalized Setting					
	CUB	AwA2	CUB			AwA2		
			U	S	H	U	S	H
CVAE-ZSL [24]	52.1	65.8	-	-	34.5	-	-	51.2
cycle-UWGAN[8]	58.6	66.8	47.9	59.3	53.0	59.6	63.4	59.8
f-CLSWGAN [52]	57.3	68.2	43.7	57.7	49.7	57.9	61.4	59.6
GF-ZSL [44]	59.6	69.2	41.5	53.3	46.7	58.3	68.1	62.8
Ours (Meta-VGAN)	**70.4**	**73.2**	55.2	48.0	**53.2**	57.4	70.5	**63.5**

Table 4: The comparison of our ZSL and GZSL result with the recent state-of-the-art generative model when using all samples. All the approach follow the same setting proposed by [51]



(a) Use all samples from seen classes to train the model



(b) Using 5 examples per seen class to train the model

Figure 2: t-NSE visualization of synthesized features on the CUB dataset for the random 10 classes. We can observe that using very few training samples the generated data distribution are very close to the case when model use all samples.

eter that minimizes the loss on the meta-train as well as the meta-validation set. The evaluation of the proposed split vs. standard split supports our claim. On the CUB and AWA2 dataset for the ZSL setup, the proposed split shows the 68.8% and 68.4%, while the standard split shows poor performance, and we obtained 66.2% and 65.7% mean per-class accuracy.

5.5. A simpler generative model trained with meta-learner for ZSL

We also compare with vanilla CVAE trained with meta-learning (a simple generative model) with the Meta-VGAN. The results are shown in Table 5, we observe that a complex model (CVAE+CGAN) synthesizes better quality features for unseen classes as compared to a simple generative model. The proposed model outperforms over CVAE based model for all datasets by a significant margin. Therefore the joint model has better generalization ability for the feature generation of the unseen classes.

6. Conclusion

We proposed a novel framework to solve ZSL/GZSL when very few samples from each of the seen classes are available during training. The various components in the proposed joint architecture of the conditional VAE and conditional GAN are augmented with meta-learners. The GAN helps to generate high-quality samples while VAE reduces the mode collapse problem that is very common in GANs. The meta-learning framework with episodic training requires only a few samples to train the complete model. The meta-learners inner loop provides a better initialization to the generative structure. Therefore, the guided discriminator provides strong feedback to the generator, and the generator can generate high-quality samples. In the ablation study, we have shown that meta-learning based training and disjoint meta-train and meta-validation classes are the crucial components of the proposed model. Extensive experiments over benchmark datasets for ZSL/GZSL show that the proposed model is significantly better than the baselines.

Datasets	Accuracy	
	5	10
SUN [53]	54.95	56.54
CUB[49]	63.70	65.63
AWA1 [51]	64.20	64.32
AwA2[51]	64.22	64.85
aPY	42.15	42.99

Table 5: ZSL results using vanilla CVAE as generative model trained with meta learner over the five standard dataset.

References

[1] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for attribute-based classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 819–826, 2013.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[3] Yuval Atzmon and Gal Chechik. Adaptive confidence smoothing for generalized zero-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[4] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Synthesized classifiers for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5327–5336, 2016.

[5] Long Chen, Hanwang Zhang, Jun Xiao, Wei Liu, and Shih-Fu Chang. Zero-shot visual recognition using semantics-preserving adversarial embedding networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[6] Zhengming Ding and Hongfu Liu. Marginalized latent semantic encoder for zero-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[7] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1785. IEEE, 2009.

[8] Rafael Felix, BG Vijay Kumar, Ian Reid, and Gustavo Carneiro. Multi-modal cycle-consistent generalized zero-shot learning. *arXiv preprint arXiv:1808.00136*, 2018.

[9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *ICML*, 2017.

[10] 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.

[11] 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*, pages 2672–2680, 2014.

[12] R Lily Hu, Caiming Xiong, and Richard Socher. Correction networks: Meta-learning for zero-shot learning. 2018.

[13] He Huang, Changhu Wang, Philip S. Yu, and Chang-Dong Wang. Generative dual adversarial network for generalized zero-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[14] Zhong Ji, Yanwei Fu, Jichang Guo, Yanwei Pang, Zhongfei Mark Zhang, et al. Stacked semantics-guided attention model for fine-grained zero-shot learning. In *Advances in Neural Information Processing Systems*, pages 5995–6004, 2018.

[15] Michael Kampffmeyer, Yinbo Chen, Xiaodan Liang, Hao Wang, Yujia Zhang, and Eric P. Xing. Rethinking knowledge graph propagation for zero-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[16] Varun Khare, Divyat Mahajan, Homanga Bharadhwaj, Vinay Kumar Verma, and Piyush Rai. A generative framework for zero shot learning with adversarial domain adaptation. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 3101–3110, 2020.

[17] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D. Yoo. Edge-labeling graph neural network for few-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

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

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

[20] 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*, 36(3):453–465, 2014.

[21] Hongyang Li, David Eigen, Samuel Dodge, Matthew Zeiler, and Xiaogang Wang. Finding task-relevant features for few-shot learning by category traversal. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[22] Jingjing Li, Mengmeng Jing, Ke Lu, Zhengming Ding, Lei Zhu, and Zi Huang. Leveraging the invariant side of generative zero-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[23] Shichen Liu, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Generalized zero-shot learning with deep calibration network. In *Advances in Neural Information Processing Systems*, pages 2005–2015, 2018.

[24] Ashish Mishra, Shiva Krishna Reddy, Anurag Mittal, and Hema A Murthy. A generative model for zero shot learning using conditional variational autoencoders. In *CVPR-Workshops*, pages 2188–2196, 2018.

[25] Ashish Mishra, Anubha Pandey, and Hema A Murthy. Zero-shot learning for action recognition using synthesized features. *Neurocomputing*, 2020.

[26] Jian Ni, Shanghang Zhang, and Haiyong Xie. Dual adversarial semantics-consistent network for generalized zero-shot learning. In *Advances in Neural Information Processing Systems*, pages 6143–6154, 2019.

[27] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S Corrado, and Jeffrey Dean. Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*, 2013.

[28] Anubha Pandey, Ashish Mishra, Vinay Kumar Verma, and Anurag Mittal. Adversarial joint-distribution learning for novel class sketch-based image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[29] Anubha Pandey, Ashish Mishra, Vinay Kumar Verma, Anurag Mittal, and Hema Murthy. Stacked adversarial network for zero-shot sketch based image retrieval. In *The*

IEEE Winter Conference on Applications of Computer Vision, pages 2540–2549, 2020.

[30] Akanksha Paul, Narayanan C. Krishnan, and Prateek Munjal. Semantically aligned bias reducing zero shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[31] Zhimao Peng, Zechao Li, Junge Zhang, Yan Li, Guo-Jun Qi, and Jinhui Tang. Few-shot image recognition with knowledge transfer. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[32] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv preprint arXiv:1909.09157*, 2019.

[33] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. *ICLR-16*, 2016.

[34] Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Few-shot learning with embedded class models and shot-free meta training. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

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

[36] Mert Bulent Sariyildiz and Ramazan Gokberk Cinbis. Gradient matching generative networks for zero-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[37] Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero- and few-shot learning via aligned variational autoencoders. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[38] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.

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

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

[41] Jie Song, Chengchao Shen, Yezhou Yang, Yang Liu, and Mingli Song. Transductive unbiased embedding for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1024–1033, 2018.

[42] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[43] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

[44] Vinay Kumar Verma, Gundeep Arora, Ashish Mishra, and Piyush Rai. Generalized zero-shot learning via synthesized examples. *CVPR*, 2018.

[45] Vinay Kumar Verma, Dhanajit Brahma, and Piyush Rai. A meta-learning framework for generalized zero-shot learning. *AAAI*, 2020.

[46] Vinay Kumar Verma, Aakansha Mishra, Ashish Mishra, and Piyush Rai. Generative model for zero-shot sketch-based image retrieval. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 704–713. IEEE, 2019.

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

[48] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

[49] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[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.

[51] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

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

[53] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492. IEEE, 2010.

[54] Guo-Sen Xie, Li Liu, Xiaobo Jin, Fan Zhu, Zheng Zhang, Jie Qin, Yazhou Yao, and Ling Shao. Attentive region embedding network for zero-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[55] Xun Xu, Timothy Hospedales, and Shaogang Gong. Transductive zero-shot action recognition by word-vector embedding. *International Journal of Computer Vision*, pages 1–25, 2017.

[56] Sasi Kiran Yelamarthi, Shiva Krishna Reddy, Ashish Mishra, and Anurag Mittal. A zero-shot framework for sketch based image retrieval. In *European conference on computer vision*, pages 316–333. Springer, 2018.

[57] Hyeonwoo Yu and Beomhee Lee. Zero-shot learning via simultaneous generating and learning. In *Advances in Neural Information Processing Systems*, pages 46–56, 2019.

[58] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3010–3019. IEEE, 2017.

[59] Pengkai Zhu, Hanxiao Wang, and Venkatesh Saligrama. Generalized zero-shot recognition based on visually semantic embedding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[60] Yizhe Zhu, Jianwen Xie, Zhiqiang Tang, Xi Peng, and Ahmed Elgammal. Semantic-guided multi-attention localization for zero-shot learning. In *Advances in Neural Information Processing Systems*, pages 14917–14927, 2019.

Supplementary: Towards Zero-Shot Learning with Fewer Seen Class Examples

Vinay Kumar Verma[1*] Ashish Mishra[2*] Anubha Pandey[2$]
Hema A. Murthy[2] Piyush Rai[1]
[1]Department of CSE, IIT Kanpur; [2]Department of CSE, IIT Madras,
{vkverma,piyush}@cse.iitk.ac.in; {mishra,hema}@cse.iitm.ac.in; anubhap93@gmail.com

Algorithm 1 Meta-VGAN for ZSL

Require: $p(\mathcal{T})$:distribution over tasks
Require: η_1, η_2: step-size hyperparameters
Randomly initialize $\theta_e, \theta_g, \theta_d$
while not done **do**
  Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$; where, $\mathcal{T}_i = \{\mathcal{T}_i^{tr}, \mathcal{T}_i^{v}\}$ such that $\mathcal{T}_i^{tr} \cap \mathcal{T}_i^{v} = \phi$
  for all $\mathcal{T}_i$ **do**
    Evaluate $\nabla_{\theta_{eg}} l^{VG}_{\mathcal{T}_i^{tr}}(\theta_{eg})$
    Evaluate $\nabla_{\theta_d} l^{D}_{\mathcal{T}_i^{tr}}(\theta_d)$
    Compute adapted parameters: $\theta^{'}_{eg} = \theta_{ed} - \eta_1 \nabla_{\theta_{eg}} l^{VG}_{\mathcal{T}_i^{tr}}(\theta_{eg})$
    Compute adapted parameters: $\theta^{'}_{d} = \theta_{d} + \eta_2 \nabla_{\theta_{d}} l^{D}_{\mathcal{T}_i^{tr}}(\theta_d)$
  Update $\theta_{eg} \leftarrow \theta_{eg} - \eta_1 \nabla_{\theta_{eg}} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} l^{VG}_{\mathcal{T}_i^{v}}(\theta^{'}_{eg})$
  Update $\theta_{d} \leftarrow \theta_{d} + \eta_2 \nabla_{\theta_{d}} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} l^{D}_{\mathcal{T}_i^{v}}(\theta^{'}_{d})$

1. Datasets Descriptions

Dataset	Attribute/Dim	#Image	Seen/Unseen Class
AwA2[3]	A/85	37322	40/10
CUB[2]	CR/1024	11788	150/50
SUN[4]	A/102	14340	645/72
aPY[1]	A/64	15339	20/12

Table 1: The benchmark datasets used in our experiments, and their statistics.

To evaluate our proposed model in comparison with several state-of-the-art ZSL and generalized ZSL methods, we applied our approach to the following benchmark ZSL datasets: SUN[4], CUB[2], AwA2[3], and aPY [1]. Table 1 shows the summary of the datasets used and their statistics.

SUN Scene Recognition: SUN is a fine-grained dataset with 717 scene categories and 14,340 images. We use the widely used split of the dataset for the ZSL setting, 645 seen classes, and 72 unseen classes. The dataset has image-level attributes. For training, we use class-level attributes obtained by combining the attributes of all the images in a class.

Animals with Attributes: AwA2 is a coarse-grained dataset with 50 classes and 37,322 images. We follow a standard zero-shot split of 40 seen (train) classes and ten unseen (test) classes. The dataset has 85-dimensional human-annotated class-attributes.

Caltech UCSD Birds 200: CUB is a fine-grained dataset with 11,788 images from 200 different types of birds, annotated with 312 attributes. We use a zero-shot split of 150 unseen and 50 seen classes. The dataset has image-level attributes like the SUN dataset. We average these image-level attributes of all the classes to obtain class attributes for training.

Attribute Pascal and Yahoo (aPY): aPY is a coarse-grained dataset with 64 attributes. The dataset has 32 classes. For Zero-Shot learning, we follow a split of 20 Pascal classes for training and 12 Yahoo classes for testing.

2. Implementation Details

Our proposed architecture Meta-VGAN has an encoder, decoder, generator, and discriminator modules, as shown in Figure-1 in the main paper. The decoder and generator modules share the same network parameters. Each of the modules has a series of FC layers followed by a ReLU and dropout layers. We concatenate image feature vector $\mathbf{x}$ extracted from ResNet-101 with class attributes vector $\mathbf{a}$ and feed to the Encoder module E. The encoder E has a series of three FC layers, and encodes the input to $\mathbf{d_z}$ (varies with datasets) dimensional latent space with mean μ and variance $\mathbf{\Sigma}$. Noise dimensions used for CUB, SUN, AwA2, and aPY datasets are 512, 20, 40, and 20, respectively. Next, we sample $\mathbf{d_z}$ dimensional noise vector from the latent space and feed it to the decoder module (or the generator module). The decoder (or generator) uses a series of 2 FC layers fol-

*Equal contribution. $ Currently author is affiliated with MasterCard

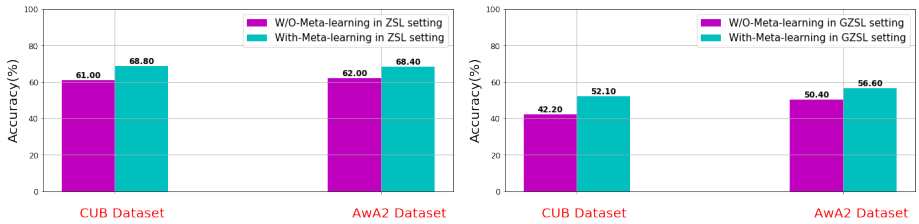


Figure 1: The left figure shows the mean per class accuracy with and without meta-learner in the ZSL setting. In the right figure the GZSL result are shown with and without meta-learning.

lowed by ReLU to generate a 2048 dimension feature vector $\hat{\mathbf{x}}$ similar to the input image feature vector $\mathbf{x}$. The generated image features $\hat{\mathbf{x}}$ are further passed through the discriminator module. The discriminator receives two types of inputs: the real image feature vector $\mathbf{x}$ that comes from the ground truth data of the training set and the synthesized image features $\hat{\mathbf{x}}$ generated by the generator module(or the decoder module). The discriminator has a series of 3 FC layers and tries to distinguish between the real image feature vector $\mathbf{x}$ and the generated image feature vector $\hat{\mathbf{x}}$. The discriminator outputs the probability of the image is real. The output value should be close to 0 for fake image features $\hat{\mathbf{x}}$, and it should be close to 1 for real image features $\mathbf{x}$.

For training, we associate each of the modules with a meta-learner agent in the Meta-VGAN model. We randomly sample 10 classes for training and ten classes for validation from the seen classes of the dataset such that they are mutually exclusive. We call this a *task*. We randomly sample 5 examples from each class of the train set and three examples from each class of the validation set. For each task, we iterate through each class of the train set multiple times and compute the adapted parameters of the network using Eq.7 and 8 in the main paper. Next, we pass the validation data through the network with initial parameters and with the computed parameters and compute the loss. We finally update the network on the validation loss, as shown in Eq.10 and 11 in the main paper. The learning rate and dropout rate used for all the datasets are 0.001 and 0.3, respectively. All the hyperparameters are selected using cross-validation.

The values of hyper-parameters η_1 and η_2, used for computation of updated parameters on training loss, are empirically chosen using a grid search in the range $[1e-1, 1e-8]$.

2.1. Comparison with and without meta-learning

Our model is a combination of CVAE and CGAN, which are integrated with a meta-learner. To illustrate the contribution of the meta-learner module, we perform an experiment when no meta-learning component is present. Figure 1 shows the comparison between with and without meta-learner in our model, for CUB and AWA2 datasets, in both standard ZSL and GZSL settings. We observe that the role of a meta-learner in our proposed model is very crucial, which helps to train our model very efficiently using a few examples per seen class. The meta-learner component boosts the model's absolute performance by 7.8% and 6.4% in standard ZSL, while by 9.9% 6.2% in GZSL for CUB and AWA2 datasets, respectively.

References

[1] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1785. IEEE, 2009.

[2] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[3] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[4] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492. IEEE, 2010.