

# Task Aligned Generative Meta-learning for Zero-shot Learning

Zhe Liu,<sup>1\*</sup> Yun Li,<sup>1\*</sup> Lina Yao,<sup>1</sup> Xianzhi Wang,<sup>2</sup> Guodong Long<sup>2</sup>

<sup>1</sup> University of New South Wales, Australia

<sup>2</sup> University of Technology Sydney, Australia

{zhe.liu1, yun.li5, lina.yao}@unsw.edu.au, {xianzhi.wang, guodong.long}@uts.edu.au

## Abstract

Zero-shot learning (ZSL) refers to the problem of learning to classify instances from novel classes (unseen) that are absent in the training set (seen). Most ZSL methods infer the correlation between visual features and attributes to train the classifier for unseen classes. They may have a strong bias towards seen classes during training. Meta-learning has been introduced to mitigate the bias, but meta-ZSL methods are inapplicable when tasks used for training are sampled from diverse distributions. In this regard, we propose a novel Task-aligned Generative Meta-learning model for Zero-shot learning (TGMZ), aiming to mitigate the potentially biased training and to enable meta-ZSL to accommodate real-world datasets that contain diverse distributions. Specifically, TGMZ incorporates an attribute-conditioned task-wise distribution alignment network that projects tasks into a unified distribution to deliver an unbiased model. Our experiments show TGMZ achieves a relative improvement of 2.1%, 3.0%, 2.5%, and 7.6% over state-of-the-art algorithms on AWA1, AWA2, CUB, and aPY datasets, respectively. Overall, TGMZ outperforms competitors by 3.6% in the generalized zero-shot learning (GZSL) setting and 7.9% in our proposed fusion-ZSL setting.

## Introduction

Most current machine learning methods focus on classifying instances into existing classes based on large amounts of labeled data (Day and Khoshgoftaar 2017). Such methods cannot well handle real-world settings where there are many classes yet insufficient instances to cover all the classes (Wang et al. 2019b). Zero-shot learning (ZSL) aims to classify novel (or unseen) classes based on the existing (or seen) training set and has achieved significant success in many fields, e.g., computer vision (Reed et al. 2016; Zhang and Saligrama 2016), natural language processing (Firat et al. 2016; Johnson et al. 2017), and human activity recognition (Wang, Miao, and Hao 2017).

Typical ZSL algorithms learn the correlation between visual features and the corresponding attributes, e.g., hand-engineering attributes (Shen et al. 2018; Liu et al. 2018b) and textual description (Srivastava, Labutov, and Mitchell

\*Equal contribution.

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

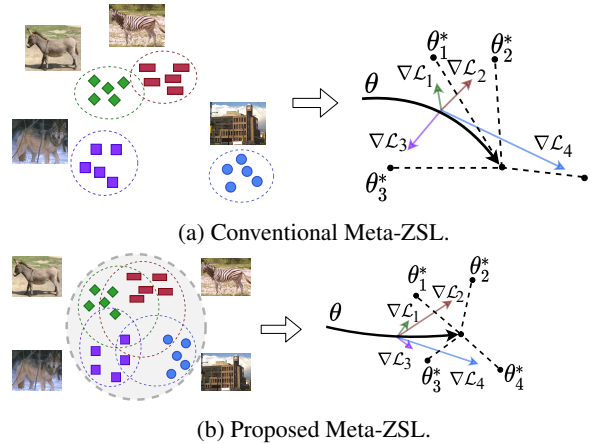


Figure 1: Visualization of model representation  $\theta$  of the conventional and the proposed meta-ZSL using 1 class for query sets. The conventional meta-ZSL (Finn, Abbeel, and Levine 2017) is biased towards class ‘building’, while the proposed aligned task distributions can avoid such local optimality.

2018; Kumar et al. 2019), and utilize the semantic information to infer the classification space for unseen classes. Among ZSL algorithms, attribute-based algorithms classify unseen classes based on visual-attribute embedding (Zhang and Koniusz 2018; Liu et al. 2020a; Wang et al. 2019c; Al-Halah and Stiefelhagen 2015), and generative methods emulate the general data distributions of unseen classes conditioned on attributes and synthesize instances for supervised training of classifiers (Mishra et al. 2018; Zhu et al. 2019; Zhang and Koniusz 2018). All the existing ZSL algorithms only optimize models based on seen classes but fail to explicitly mimic ZSL settings that transfer knowledge from seen classes to unseen classes at the training time. Consequently, these algorithms are biased towards existing visual-attribute correlation and fail to speculate the real classification space for unseen classes.

To address the above issue, some work (Qin et al. 2019; Pal and Balasubramanian 2019; Soh, Cho, and Cho 2020; Demertzis and Iliadis 2020; Nooralahzadeh et al. 2020; Verma, Brahma, and Rai 2020) introduces meta-learning, e.g., model-agnostic meta-learning frame-

work (Finn, Abbeel, and Levine 2017), into ZSL, namely meta-ZSL. Meta-ZSL splits existing training classes into two disjoint sets, namely support and query sets, to mimic seen and unseen classes. Then, it randomly picks up classes from support and query sets to construct different tasks for training. This way, meta-ZSL can learn to adapt from seen classes to unseen classes explicitly and thus obtain an unbiased model (Snell, Swersky, and Zemel 2017; Vinyals et al. 2016).

However, current meta-ZSL approaches (Verma, Brahma, and Rai 2020; Demertzis and Iliadis 2020; Soh, Cho, and Cho 2020) directly integrate meta-learning and ZSL without considering the limitations posed by diverse data distributions in ZSL. Thus, the learned models may be misguided towards extremely different distributions. Take ‘wolf’, ‘donkey’, ‘zebra’, and ‘building’ from the real-world dataset, aPY (Farhadi et al. 2009) (shown in Figure 1), for example. The first three classes (‘wolf’, ‘donkey’, ‘zebra’) are animals and are dissimilar to the class ‘building’. Suppose the four classes are the query sets in four tasks. Conventional meta-ZSL methods optimize the model representation  $\theta$  based on the largest component of the overall gradient of the four tasks, i.e.,  $\nabla \mathcal{L}_4$ . Thus, the model representation  $\theta$  will be biased towards the optimal solution to ‘building’ ( $\theta_4^*$ ) and become less discriminative on animal classes. Therefore, it is necessary to align task distributions in meta-ZSL to enable models to learn each class more moderately and robustly, as illustrated in Figure 1 (b).

In this paper, we propose a novel Task-aligned Generative Meta-learning model for Zero-shot learning (TGMZ). TGMZ uses attribute-conditioned Task adversarial AutoEncoder (TAE) to align distributions on multiple random tasks with attribute side information. The TAE extracts visual and attribute characteristics from original instances by encoding data into aligned embedding in a unified distribution. Then, a Meta conditional Generative Adversarial Network (MGAN) simulates the unbiased distribution for unseen classes. Each module in MGAN is modified with a meta-learning agent, which optimizes model parameters. To prove the superiority of our idea in handling diverse task distributions, we evaluate our model in the ZSL setting and two more challenging settings: generalized zero-shot learning (GZSL) and our proposed fusion-ZSL setting. GZSL evaluates models on both seen and unseen classes, and the fusion-ZSL setting evaluates models on the fused datasets. Our contributions in this work are summarized as follows:

- We propose a novel task-wise alignment generative meta-model, i.e., TGMZ, for zero-shot learning. TGMZ uses attribute-conditioned TAE to align task-wise distributions and adopts MGAN to learn an unbiased model for classifying instances for unseen classes to overcome the potential distribution disjointedness in meta-ZSL.
- We carry out extensive ZSL and GZSL experiments on four benchmark datasets. The results exhibit that our model significantly outperforms state-of-the-art algorithms, demonstrating the superiority of TGMZ.
- We evaluate our model the effectiveness of TGMZ in handling diverse task distributions under a novel fusion-ZSL

setting (i.e., combined dataset experiments). The embedding spaces of the synthetic instances of our model are more discriminative than state-of-the-art algorithms on both single and combined datasets, demonstrating the effectiveness of task distribution alignment.

## Related Work

### Zero-shot Learning

The current ZSL algorithms fall into two main categories (Mishra et al. 2018): attribute-based ZSL and generative ZSL. Attribute-based algorithms aim to learn the mapping from visual space to the semantic space. They project instances of unseen classes to attribute embedding and then predict their class labels by finding the most similar class attribute (Romera-Paredes and Torr 2015; Xian et al. 2016; Zhang, Xiang, and Gong 2017; Liu et al. 2018a). For example, Kodirov et al. (Kodirov, Xiang, and Gong 2017) propose to apply an encoder-decoder structure to extract more feature information supervised by reconstruction loss. Changpinyo et al. (Changpinyo et al. 2016) propose to align the semantic space and image space, thus extracting more semantic information in the embedding. Zhang et al. (Zhang and Koniusz 2018) propose a well-established kernel-based method with orthogonality constraints to better learn the non-linear mapping relationship. Generative ZSL methods classify unseen classes based on synthesizing instances according to the attribute information (Verma and Rai 2017; Chen et al. 2018b; Zhu et al. 2018; Mishra et al. 2018). For example, Zhu et al. (Zhu et al. 2018) and Xian et al. (Xian et al. 2018) apply a generative adversarial network (Goodfellow et al. 2014) with an auxiliary classifier to regularize the generator to carry correct class information based on Fully Connected Networks (FCNs) and Convolutional Neural Networks (CNNs). Kumar Verma et al. (Kumar Verma et al. 2018) adopt conditional autoencoder (Sohn, Lee, and Yan 2015) enhanced with a multivariate regressor to achieve generalized synthetic instances. Zhu et al. (Zhu et al. 2019) further propose to optimize the conditional autoencoder with an altering propagation using maximum likelihood estimation. The conventional ZSLs only consider the visual-attribute correlation of existing classes and tend to be biased towards the known attribute mapping or data distribution.

To better mimic the ZSL setting, previous studies (Wang et al. 2019c; Verma, Brahma, and Rai 2020; Qin et al. 2019; Nooralahzadeh et al. 2020) introduce meta-learning (Finn, Abbeel, and Levine 2017) into ZSL to make the model more suitable for transferring knowledge from seen classes to unseen classes. For example, Wang et al. (Wang et al. 2019c) fuse meta-learning and attribute-based ZSL method to map visual features to task-aware embedding. Soh et al. (Soh, Cho, and Cho 2020) combine meta-learning with CNNs and utilize a single gradient update to obtain a generic initialization suitable for internal learning. Verma et al. (Verma, Brahma, and Rai 2020) first introduce meta-learning-based generative ZSL and apply meta-learner on each module of generative ZSL. All the above methods combine meta-learning with ZSL directly while neglecting the bias of meta-learning caused by the diversity of task distributions.

## Domain Alignment

Most current studies on domain adaptation handle different task distributions via single-source source-to-target alignment (Wilson and Cook 2020; Chen et al. 2018a; Wang, Michau, and Fink 2020; Sankaranarayanan et al. 2018; Liu et al. 2020b). For example, Gholami et al. (Wang, Michau, and Fink 2020) use an adversarial autoencoder and a discriminative discrepancy loss function to align two domains. Guo et al. (Guo, Pasunuru, and Bansal 2020) extend single-source domain adaptation to multiple sources based on the distance discrepancy. Zhao et al. (Zhao et al. 2019) propose end-to-end adversarial domain adaptation for multiple sources based on a pixel-level cycle-consistency loss. Wang et al. (Wang et al. 2019a) enhance multi-source adversarial alignment by introducing task-specific decision boundaries. The aforementioned work has not considered aligning domains using the attribute side information on multiple sources without a fixed source-to-target relationship.

## Summary

Compared with the related work, our contributions are three-fold. First, we propose task-wise TAE, which extracts both class and visual feature information for reconstruction, to carry out task-wise distribution alignment. Second, we fuse meta-learning and ZSL in a robust way. TAE uses attribute side information to provide aligned embedding for ZSL and to prevent meta-learner from optimizing the model to be biased due to disjoint task distributions. Third and the last, differing from previous work that directly learns visual features for classification, our model learns unbiased synthesized embedding, which is more suitable for novel classes.

## Methodology

### Problem Definition and Overview

Suppose dataset  $\mathcal{D} = \{X, A, Y\}$  contains visual features  $X$ , attribute vectors  $A$ , and class labels  $Y$ , and lowercase letters  $x, a, y$  are instances from the respective sets.  $\mathcal{D}$  contains two disjoint subsets: training set  $\mathcal{D}_{tr}$  and testing set  $\mathcal{D}_{ts}$ . The goal of ZSL is to transfer knowledge from  $\mathcal{D}_{tr}$  to  $\mathcal{D}_{ts}$ . Our model aims to emulate the attribute-conditioned data distribution to synthesize unseen classes’ instances and then train a classifier supervised by the synthetic instances to predict real instances from unseen classes.

Different from conventional ZSL settings, we divide the training set  $\mathcal{D}_{tr}$  into a support set  $\mathcal{D}_{sup}$  and a disjoint query set  $\mathcal{D}_{qry}$  to mimic seen and unseen classes, respectively, during training. To carry out episode-wise training, we sample tasks  $\{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_i\}$  following the N-way K-shot setting in (Verma, Brahma, and Rai 2020). Given an arbitrary task  $\mathcal{T}_i = \{\mathcal{D}_{sup}^{\mathcal{T}_i}, \mathcal{D}_{qry}^{\mathcal{T}_i}\}$ , we sample  $K$  classes with  $N$  images for each class from  $\mathcal{D}_{sup}$  and  $\mathcal{D}_{qry}$ , respectively. Our method is inductive (Xian et al. 2019) so that no extra information from  $\mathcal{D}_{ts}$  is used in the training.

Our training procedure consists of two phases: task distribution alignment and task-aligned generative zero-shot learning. The former phase aligns the diverse sampled tasks

to a unified task distribution to ease the biased optimization. The latter phase synthesizes instances in the aligned distribution conditioned on attribute vectors and uses meta-learner to train the model. We illustrate the details of the two phases in the following subsections.

### Task Distribution Alignment

In this phase, we regularize task distributions into a unified distribution along with the training of TGMZ. We consider three requirements for TAE: (i) the encoder of TAE should be capable of aligning task-wise distributions; (ii) TAE should regularize encoder based on the visual-attribute correlation; (iii) the embedding should extract visual and class characteristics from the original features.

TGMZ relies on episode-wise training (Wang et al. 2019c; Verma, Brahma, and Rai 2020) to jointly handle multiple tasks in each iteration while optimizing the model. The first requirement enables the encoder to align multiple sampled task distributions synchronously and to be optimized along with generative networks in the second phase. The second requirement enables TAE to utilize attribute side information during regularization, thus enriching the attribute information in the embedding to better fits the ZSL setting. The third requirement ensures optimizing embedding based on the original visual and class information, which can increase the authenticity of the synthetic instances.

Our proposed TAE consists of three components: Task Encoder  $q_{\theta_{te}}(e|x)$ , Task Decoder  $p_{\theta_{td}}(x|e)$ , Task Discriminator  $f_{\theta_{tdis}}(e, a) \rightarrow m$ , and Task Classifier  $f_{\theta_{cls}}(e) \rightarrow y$ , where  $e$  denotes the encoded visual features, i.e., aligned embedding for task distributions;  $x, a, y$  denote the original visual features, attribute vectors and class labels;  $\theta$  denotes the parameter for the corresponding module. The task encoder aims to project tasks into an embedding  $e$  that follows a unified distribution; the task decoder helps reconstruct original instances; task classifier aims to predict class labels. Specifically, we adopt a multi-label classifier as the discriminator  $f_{\theta_{tdis}}(e, a) \rightarrow m$  for TAE and assign pseudo-labels  $m$  for tasks to differentiate task distributions, i.e.,  $\forall \mathcal{T}_i, m_{\mathcal{T}_i} = i$ .

For each task  $\mathcal{T}_i$ , we design loss functions  $\mathcal{L}_{tdis}$  and  $\mathcal{L}_{tc}$  to optimize  $\theta_{tdis}$  and other modules  $\theta_{tc}$ , respectively:

$$\min_{\theta_{tc}} \mathcal{L}_{tc} = \mathbb{E}_{e_i \sim q_{\theta_{te}}(e_i|x_i)} [-\log p_{\theta_{td}}(x_i|e_i)] - \mathcal{L}_{ce}(f_{\theta_{tdis}}(e_i, a_i), m_i) + \mathcal{L}_{ce}(f_{\theta_{cls}}(e_i), y_i) \quad (1)$$

$$\max_{\theta_{tdis}} \mathcal{L}_{tdis} = -\mathcal{L}_{ce}(f_{\theta_{tdis}}(e_i, a_i), m_i) \quad (2)$$

*s.t.*  $x_i \sim \mathcal{T}_i, \quad \theta_{tc} = \{\theta_{te}, \theta_{td}, \theta_{cls}\}$

where  $\mathcal{T}_i$  denotes a sampled task;  $x_i$  is an arbitrary sample from  $\mathcal{T}_i$ ;  $e_i, a_i, m_i, y_i$  are the corresponding encoded embedding, attribute vector, pseudo-label, and ground-truth class label for instance  $x_i$ , respectively;  $\mathcal{L}_{ce}$  denotes the Cross Entropy loss function.

The second item in  $\mathcal{L}_{tc}$  and the loss  $\mathcal{L}_{tdis}$  construct a two-player minimax game between the task discriminator and the task encoder. They jointly enable the model to meet the first two requirements. On the one hand, the task encoder learns

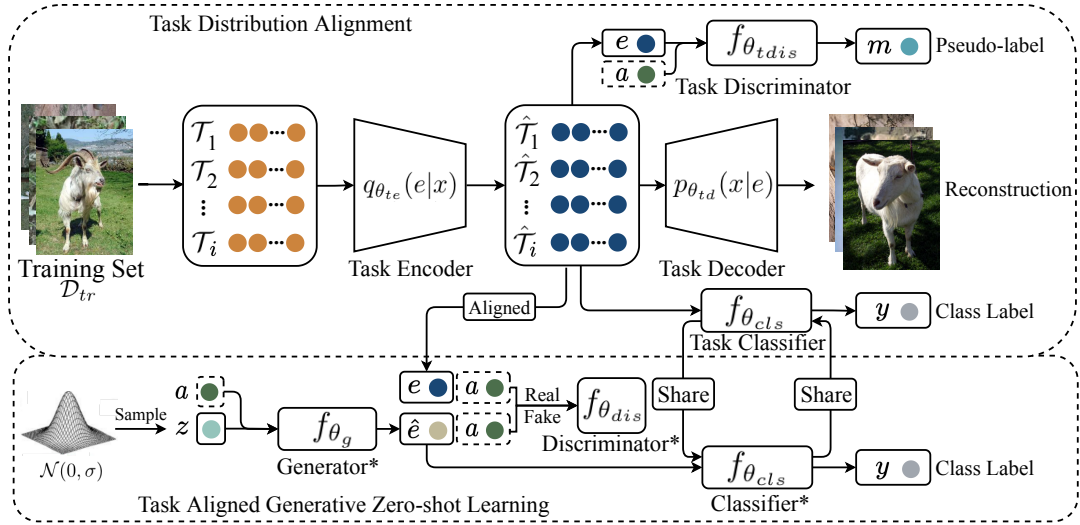


Figure 2: Architecture of the proposed TGMZ.  $a$  is the attribute vector. Discriminator\*, Generator\* and Classifier\* denote the modules that are updated by meta-learner.

to encode diverse tasks into the embedding to confuse the multi-label discriminator with the pseudo-labels  $m$ ; on the other hand, the task discriminator attempts to distinguish different task distributions. After such adversarial training, the task encoder will learn to align multiple diverse tasks into a unified distribution to fool task discriminator—the tasks that share the similar distribution will easily confuse task discriminator, and the unique tasks that follow the different distributions will be aligned into a unified distribution, making TAE satisfy the first requirement. Also, the alignment within tasks can also be carried out when the tasks contain multiple classes. Since task encoder learns to make different tasks be similar distributions, we can construct different tasks to align the classes within tasks. For example, given four classes  $(a, b, c, d)$ , if each task contains two classes, we may construct task pairs for episodes by  $\{(a, b), (c, d)\}$ , and  $\{(a, c), (b, d)\}$ . During training, task alignment module will learn to align distributions  $(a, b)$  to  $(c, d)$ ,  $(a, c)$  to  $(b, d)$ . Finally, after enough episodes,  $(a, b, c, d)$  will be aligned in a unified distribution.

Besides, the attribute-conditioned discriminator distinguishes synthetic instances based on attribute vectors. Since attribute vectors are invisible to the task encoder, the task encoder will be regularized to infer the attribute information of classes from visual features to meet the second requirement. The other two terms in  $\mathcal{L}_{tc}$  are the reconstruction loss and auxiliary classification loss. The former forces the encoder-decoder to reconstruct original visual features, while the latter makes the task encoder extract class information during the encoding, which fulfills the third requirement.

We summarize the episodic loss function  $\mathcal{L}_{align}$  for task alignment on multiple tasks  $\{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_i\}$  as follows:

$$\min_{\theta_{tc}} \max_{\theta_{tdis}} \sum_{j=1}^i \mathbb{E}_{e_j \sim q_{\theta_{te}}(e_j|x_j), x_j \sim \mathcal{T}_j} [-\log p_{\theta_{td}}(x_j|e_j)] \quad (3)$$

$$-\mathcal{L}_{ce}(f_{\theta_{tdis}}(e_j, a_j), m_j) + \mathcal{L}_{ce}(f_{\theta_{cls}}(e_j), y_j)$$

## Task Aligned Generative Zero-shot Learning

We learn a general data distribution to generate instances for unseen classes using MGAN. Specifically, we adopt the encoded embedding of original tasks, i.e., aligned tasks, as input for MGAN. Let  $\hat{\mathcal{T}}_i = \{\hat{\mathcal{T}}_i^{sup}, \hat{\mathcal{T}}_i^{qry}\}$  be the encoded embedding of  $\mathcal{T}_i$  and the subsets,  $\hat{\mathcal{T}}_i^{sup}$  and  $\hat{\mathcal{T}}_i^{qry}$ , be the embedding of  $\mathcal{D}_{sup}^{\mathcal{T}_i}, \mathcal{D}_{qry}^{\mathcal{T}_i}$ , respectively. Suppose the aligned tasks follows a unified distribution  $\hat{\mathcal{T}}_i \sim p(\hat{\mathcal{T}})$ , and  $p(\hat{\mathcal{T}})$  represents the task distribution over the support and query sets. For MGAN, we apply a Generator  $f_{\theta_g}(z, a) \rightarrow \hat{e}$ , a Discriminator  $f_{\theta_{dis}}(e, a) \rightarrow [0, 1]$  and a Classifier  $f_{\theta_{cls}}(e) \rightarrow y$ , where  $z$  is the random noise from a normal distribution and  $\hat{e}$  denotes synthetic instances. The discriminator predicts real instances as 1 and fake instances as 0. The classifier in MGAN shares the same weights as the task classifier in TAE, which provides a warm-start initialization for training (we use the same notation for the classifiers). Each module of MGAN is integrated with a meta-learner, and the training is a two-step procedure based on gradient descent optimization.

MGAN follows a two-step procedure. First, meta-learner computes task-specific optimal parameters based on the support sets, i.e.,  $\hat{\mathcal{T}}_i^{sup}$  without updating model parameters  $\theta$ . Differing from conventional meta-learning, we seek the overall optimal parameters for all the tasks in support sets rather than searching a set of optimal model parameters for each task (Verma, Brahma, and Rai 2020) to achieve better stability in optimizing generative models. Second, meta-learner computes the gradients of optimizing the overall optimal parameters towards query sets, i.e.,  $\hat{\mathcal{T}}_i^{qry}$ , and then summarizes the gradients to update model parameters  $\theta$ ; this enables the model to learn transferable parameters from seen classes to unseen classes.

The training relies on the task-specific loss function  $\mathcal{L}_{zsl}$  to compute gradients on the support and query sets for ob-

---

**Algorithm 1** TGMZ Training Procedure

---

**Require:**  $\mathcal{D}_{tr}$ : training dataset**Require:**  $lr_{tc}, lr_{tdis}$ : learning rates**Require:**  $\alpha_1, \alpha_2, \beta_1, \beta_2$ : step sizes

- 1: Initialize  $\theta_{tc}, \theta_{tdis}, \theta_{gc}, \theta_{dis}$
  - 2: Split  $\mathcal{D}_{tr}$  into disjoint subsets  $\mathcal{D}_{sup}$  and  $\mathcal{D}_{qry}$
  - 3: **while** not done **do**
  - 4:   Sample tasks  $\{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_i\}$  from  $\mathcal{D}_{sup}$  and  $\mathcal{D}_{qry}$
  - 5:   Update  $\theta_{tdis} \leftarrow Adam(\mathcal{L}_{tdis}, lr_{tdis})$
  - 6:   Update  $\theta_{tc} \leftarrow Adam(\mathcal{L}_{tc}, lr_{tc})$
  - 7:   **for**  $j \in [1, i]$  **do**
  - 8:      $\hat{\mathcal{T}}_j \leftarrow f_{\theta_{tc}}(\mathcal{T}_j)$
  - 9:     Evaluate  $\nabla_{\theta_{dis}} \mathcal{L}_{\hat{\mathcal{T}}_j^{sup}}^{zsl}(\theta_{dis})$  w.r.t  $\hat{\mathcal{T}}_j^{sup} \in \hat{\mathcal{T}}_j$
  - 10:     Update  $\theta'_{dis} = \theta_{dis} + \alpha_1 \nabla_{\theta_{dis}} \mathcal{L}_{\hat{\mathcal{T}}_j^{sup}}^{zsl}(\theta_{dis})$
  - 11:     Evaluate  $\nabla_{\theta_{gc}} \mathcal{L}_{\hat{\mathcal{T}}_j^{sup}}^{zsl}(\theta_{gc})$  w.r.t  $\hat{\mathcal{T}}_j^{sup} \in \hat{\mathcal{T}}_j$
  - 12:     Update  $\theta'_{gc} = \theta_{gc} - \alpha_2 \nabla_{\theta_{gc}} \mathcal{L}_{\hat{\mathcal{T}}_j^{sup}}^{zsl}(\theta_{gc})$
  - 13:     Update  $\theta_{dis} \leftarrow \theta_{dis} + \beta_1 \sum \hat{\mathcal{T}}_j \nabla_{\theta_{dis}} \mathcal{L}_{\hat{\mathcal{T}}_j^{sup}}^{zsl}(\theta'_{dis})$
  - 14:     Update  $\theta_{gc} \leftarrow \theta_{gc} - \beta_2 \sum \hat{\mathcal{T}}_j \nabla_{\theta_{gc}} \mathcal{L}_{\hat{\mathcal{T}}_j^{sup}}^{zsl}(\theta'_{gc})$
- 

taining the overall optimal parameters. Therefore, we start by introducing the task-specific loss function  $\mathcal{L}_{zsl}$  as follows:

$$\begin{aligned} \min_{\theta_{gc}} \max_{\theta_{dis}} \mathbb{E}_{\hat{\mathcal{T}}_i} [f_{\theta_{dis}}(e_i, a_i)] - \mathbb{E}_{a_i, z \sim \mathcal{N}(0, \sigma)} [f_{\theta_{dis}}(\hat{e}_i, a_i)] \\ + \mathcal{L}_{ce}(f_{\theta_{dis}}(\hat{e}_i), y_i) \\ \text{s.t. } \hat{e}_i \sim f_{\theta_g}(z, a_i), \quad \theta_{gc} = \{\theta_g, \theta_{cls}\} \end{aligned} \quad (4)$$

where  $\hat{e}_i$  is a synthetic instance conditioned on  $a_i$ ;  $z$  is the random noise from Normal Distribution  $\mathcal{N}(0, \sigma)$ ;  $\mathcal{L}_{ce}$  is the Cross Entropy loss function.

We optimize the task-specific loss function in an adversarial manner. In each episode,  $\mathcal{L}_{zsl}$  first optimizes the discriminator  $\theta_{dis}$  based on the first two items to enable the discriminator to distinguish real and synthetic instances precisely. Then,  $\mathcal{L}_{zsl}$  optimizes the generator and the classifier, i.e.,  $\theta_{gc}$ , to synthesize instances with class information and to confuse attribute-conditioned discriminator according to the last two items. Finally, the generator emulates the real data distribution based on attribute vectors for unseen classes.

Let  $\nabla_{\theta_{dis}} \mathcal{L}_{\hat{\mathcal{T}}_i}^{zsl}(\theta_{dis})$  be the gradient of  $\mathcal{L}_{zsl}$  for module  $\theta_{dis}$  (subscript of  $\nabla$ ) conditioned on parameter  $\theta_{dis}$  (in brackets) and data  $\hat{\mathcal{T}}_i$ . Similarly, we signify  $\nabla_{\theta_{gc}} \mathcal{L}_{\hat{\mathcal{T}}_i}^{zsl}(\theta_{gc})$  as the gradient for  $\theta_{gc}$  (under the same condition). On this basis, we sort out the process of finding the overall task-specific optimal parameters for  $\hat{\mathcal{T}}$  as follows:

$$\theta'_{dis} = \theta_{dis} + \alpha_1 \nabla_{\theta_{dis}} \sum_{\hat{\mathcal{T}}_j \sim p(\hat{\mathcal{T}})} \mathcal{L}_{\hat{\mathcal{T}}_j^{sup}}^{zsl}(\theta_{dis}) \quad (5)$$

$$\theta'_{gc} = \theta_{gc} - \alpha_2 \nabla_{\theta_{gc}} \sum_{\hat{\mathcal{T}}_j \sim p(\hat{\mathcal{T}})} \mathcal{L}_{\hat{\mathcal{T}}_j^{sup}}^{zsl}(\theta_{gc}) \quad (6)$$

where  $\alpha_1, \alpha_2$  denote the step sizes for the optimization.

Dataset	#Attribute Dim	#Image	#Seen/Unseen
AWA1	85	30,475	40/10
AWA2	85	37,322	40/10
CUB	1024	11,788	150/50
aPY	64	15,339	20/12

Table 1: Dataset statistics. # denotes number.

With optimal parameters for seen classes of sampled tasks, the meta-learner updates each module as follows:

$$\theta_{dis} \leftarrow \theta_{dis} + \beta_1 \nabla_{\theta_{dis}} \sum_{\hat{\mathcal{T}}_j \sim p(\hat{\mathcal{T}})} \mathcal{L}_{\hat{\mathcal{T}}_j^{qry}}^{zsl}(\theta'_{dis}) \quad (7)$$

$$\theta'_{gc} \leftarrow \theta_{gc} - \beta_2 \nabla_{\theta_{gc}} \sum_{\hat{\mathcal{T}}_j \sim p(\hat{\mathcal{T}})} \mathcal{L}_{\hat{\mathcal{T}}_j^{qry}}^{zsl}(\theta'_{gc}) \quad (8)$$

where  $\beta_1, \beta_2$  denote step sizes for transfer learning from seen classes to unseen classes.

Eq. 7 and Eq. 8 illustrate the episode-wise optimization for TGMZ, with the detailed algorithm procedure described in Algorithm 1.

## Experiment

### Experiment Setup

We conduct extensive experiments on four benchmark datasets: AWA1 (Lampert, Nickisch, and Harmeling 2009), AWA2 (Xian et al. 2019), CUB (Welinder et al. 2010), and aPY (Farhadi et al. 2009). AWA1, AWA2, and CUB are animal datasets. In particular, CUB consists of fine-grained bird species that are hard to discriminate; aPY comprises highly diverse classes, e.g., buildings and animals. We use hand-engineering attribute vectors in AWA1, AWA2, and aPY, and use 1024-dimensional embedding attributes extracted by Reed et al. (Reed et al. 2016) in the CUB dataset, which shows superior performance than the original attributes. We divide the datasets into seen and unseen classes following the proposed split (PS) (Xian et al. 2019) and adopt visual features from pre-trained ResNet-101, according to Xian et al. (Xian et al. 2019). The dataset statistics and train/test split are shown in Table 1.

We compare our method with fifteen state-of-the-art algorithms in ZSL and GZSL and four representative algorithms in fusion-ZSL. Note that we provide the reproduced ZSML, GZSL, ABP, and DEM results in our experiments. In the ZSL and fusion-ZSL settings, we evaluate our model using Linear-SVM (Verma, Brahma, and Rai 2020) and Softmax (i.e., two fully connected layers followed by batch normalization). Since SVM is time-consuming, we only use Softmax as the classifier to evaluate our model in the GZSL setting. More details about *Dataset Description, Model Architecture, Parameter Setting* and *Convergence Analysis* can be found in **Supplementary Material**.

### Zero-shot Learning

In the ZSL setting, we evaluate our model using linear-SVM and Softmax (Table 2). Compared with extensive state-of-the-art algorithms, our model achieves 2.1%, 3.0%, 2.5%

Method	AWA2	AWA1	CUB	aPY
$\psi$ ESZSL (Romera-Paredes and Torr 2015)	58.6	58.2	53.9	38.3
$\psi$ LATEM (Xian et al. 2016)	55.8	55.1	49.3	35.2
$\psi$ SYNC (Changpinoy et al. 2016)	46.6	54.0	55.6	23.9
$\psi$ DEM (Zhang, Xiang, and Gong 2017)	67.1	68.4	51.7	35.0
SAE (Kodirov, Xiang, and Gong 2017)	54.1	53.0	33.3	24.1
Gaussian-Kernal (Zhang and Koniusz 2018)	61.6	60.5	52.2	38.9
*TAFE-Net (Wang et al. 2019c)	69.3	70.8	56.9	42.2
APNet (Liu et al. 2020a)	68.0	68.0	57.7	41.3
$\psi$ GFZSL (Verma and Rai 2017)	67.0	69.4	49.2	38.4
$\psi$ SP-AEN (Chen et al. 2018b)	58.5	-	55.4	24.1
GAZSL (Zhu et al. 2018)	70.2	68.2	55.8	41.3
$\psi$ SE-GZSL (Kumar Verma et al. 2018)	69.2	69.5	59.6	-
ABP (Zhu et al. 2019)	70.4	69.3	58.5	-
ZVAE (Gao et al. 2020)	69.3	71.4	54.8	37.4
$\psi$ *ZSML (Verma, Brahma, and Rai 2020)	76.1	73.5	68.3	35.0
$\psi$ *TGMZ-SVM	73.2	70.9	<b>70.0</b>	44.6
$\psi$ *TGMZ-Softmax	<b>78.4</b>	<b>75.1</b>	66.1	<b>45.4</b>

Table 2: ZSL average per-class Top-1 accuracy results. Attribute-based methods are shown at the top and generative methods are at the bottom. \* denotes meta-ZSL method.  $\psi$  denotes using CNN-RNN feature for CUB dataset.

and 7.6% relative improvements on AWA1, AWA2, CUB, and aPY, respectively. Both SVM and Softmax can achieve state-of-the-art performance, with TGMZ-Softmax outperforming the other algorithms on three datasets and TGMZ-SVM exhibiting the best performance on the CUB dataset. The results show that Softmax can well fit diverse classes while SVM is more suitable for classifying similar classes.

Meta-ZSLs, including TAFE-Net, ZSML, and TGMZ, achieve the best performance among the algorithms in the same categories, indicating the effectiveness of incorporating meta-learning. Although most generative methods obtain worse performance than attribute-based methods on aPY, TGMZ’s outperformance demonstrates the advantages of TAE in generative methods.

### Fusion Zero-shot Learning

In this experiment, we use the fused dataset to validate the model’s ability to handle datasets with diverse task distributions in the ZSL settings: AWA1&aPY, AWA2&aPY, and AWA1&AWA2&aPY (AWA&aPY). Figure 3 shows the average per-class Top-1 accuracy results in the fusion-ZSL setting. Our model achieves the best performance among the compared algorithms, achieving 5.5%, 7.9%, and 3.6% improvement on AWA1&aPY, AWA2&aPY, and AWA&aPY, respectively. The improvement demonstrates TGMZ’s capability to handle diverse task distributions and the advantages of the task alignment.

### Generalized Zero-shot Learning

We report our model’s performance on four datasets using three evaluation metrics (Xian et al. 2019), namely average per-class Top-1 accuracy for unseen (U), seen (S), and the harmonic mean (H), where  $H = \frac{2*U*S}{U+S}$ . Compared with state-of-the-art results, our model yields 3.7%, 2.8%, and 2.5% improvement in harmonic mean score on AWA2, CUB, and aPY, respectively. Also, our model obtains the

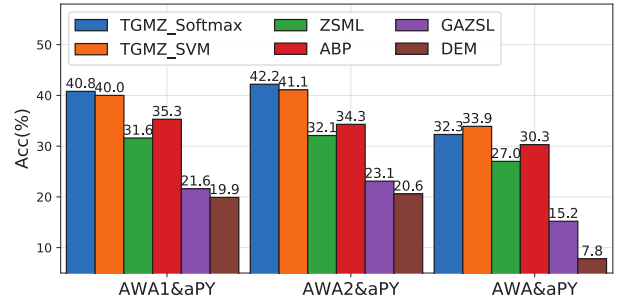


Figure 3: Fusion-ZSL average per-class Top-1 accuracies.

best performance on the unseen classes of AWA1, AWA2, and CUB. With respect to GZSL results, our model can effectively infer the potential visual-attribute correlation for unseen classes and prevent being biased towards the seen correlation. Overall, our model exhibits consistent performance improvement in different settings (as shown in Table 2, Table 3, and Figure 3), demonstrating the robustness and the superiority of TGMZ.

### Synthetic Feature Embedding Analysis

Figure 4 provides a visualization of the synthetic features of ZSML and our model on AWA2 and the combined AWA&aPY dataset. The original features are projected with t-Distributed Stochastic Neighbor Embedding (t-SNE). ZSML can only generate discriminative distributions on a few classes in AWA&aPY, indicating it is biased towards some classes. In contrast, our model can synthesize more discriminative feature space than ZSML on both AWA2 and AWA&aPY. The discriminative embedding space demonstrates the effectiveness of our method in preventing meta-learner from being biased towards certain classes.

Method	AWA2			AWA1			CUB			aPY		
	U	S	H	U	S	H	U	S	H	U	S	H
$\psi$ ESZSL (Romera-Paredes and Torr 2015)	5.9	77.8	11.0	6.6	75.6	12.1	12.6	63.8	21.0	2.4	70.1	4.6
$\psi$ LATEM (Xian et al. 2016)	11.5	77.3	20.0	7.3	71.7	13.3	15.2	57.3	24.0	0.1	73.0	0.2
$\psi$ SYNC (Changpinyo et al. 2016)	10.0	90.5	18.0	8.9	<b>87.3</b>	16.2	11.5	<b>70.9</b>	19.8	7.4	66.3	13.3
$\psi$ DEM (Zhang, Xiang, and Gong 2017)	30.5	86.4	45.1	32.8	84.7	47.3	19.6	57.9	29.2	11.1	75.1	19.4
SAE (Kodirov, Xiang, and Gong 2017)	1.1	82.2	2.2	1.8	77.1	3.5	7.8	54.0	13.6	0.4	<b>80.9</b>	0.9
Gaussian-Kernal (Zhang and Koniusz 2018)	18.9	82.7	30.8	17.9	82.2	29.4	21.6	52.8	30.6	10.5	76.2	18.5
*TAFE-Net (Wang et al. 2019c)	36.7	<b>90.6</b>	52.2	50.5	84.4	63.2	41.0	61.4	49.2	24.3	75.4	36.8
APNet (Liu et al. 2020a)	54.8	83.9	66.4	59.7	76.6	67.1	48.1	59.7	51.7	32.7	74.7	45.5
$\psi$ f-CLSWGAN (Xian et al. 2018)	57.9	61.4	59.6	61.4	57.9	59.6	43.7	57.7	49.7	-	-	-
GAZSL (Zhu et al. 2018)	35.4	86.9	50.3	29.6	84.2	43.8	31.7	61.3	41.8	14.2	78.6	24.0
$\psi$ SE-GZSL (Kumar Verma et al. 2018)	58.3	68.1	62.8	56.3	67.8	61.5	41.5	53.3	46.7	-	-	-
GDAN (Huang et al. 2019)	32.1	67.5	43.5	-	-	-	39.3	66.7	49.5	30.4	75.0	43.4
ABP (Zhu et al. 2019)	55.3	72.6	62.6	57.3	67.1	61.8	47.0	54.8	50.6	-	-	-
ZVAE (Gao et al. 2020)	57.1	70.9	62.5	58.2	66.8	62.3	43.6	47.9	45.5	32.0	52.2	39.7
$\psi$ *ZSML (Verma, Brahma, and Rai 2020)	58.9	74.6	65.8	57.4	71.1	63.5	60.0	52.1	55.7	<b>36.3</b>	46.6	40.9
$\psi$ *TGMZ	<b>64.1</b>	77.3	<b>70.1</b>	<b>65.1</b>	69.4	<b>67.2</b>	<b>60.3</b>	56.8	<b>58.5</b>	34.8	77.1	<b>48.0</b>

Table 3: GZSL average per-class Top-1 accuracy results. \* denotes meta-ZSL method. U and S represent the accuracy score for seen and unseen classes, respectively. H denotes the harmonic mean of U and S.  $\psi$  denotes using CNN-RNN feature for CUB.

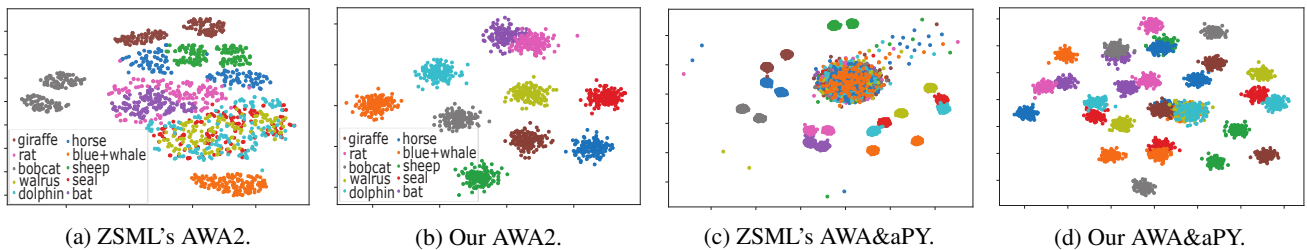


Figure 4: Synthetic feature embedding analysis.

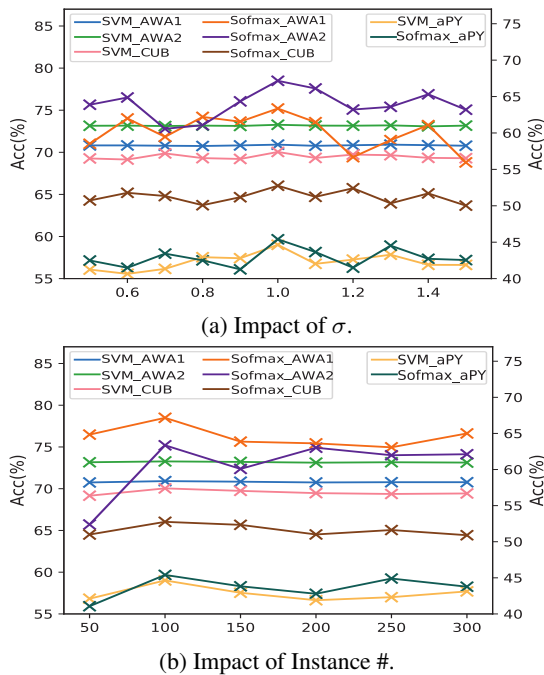


Figure 5: Hyper-parameter analysis.

## Hyper-parameter Ablation Study

We carry out an ablation study for  $\sigma$  and the number of synthetic instances. By default, we set  $\sigma = 1$  and the sample number to 100. The ZSL results of SVM and Softmax on four datasets (Figure 5) shows the parameter selection merely affect SVM, while Softmax is only slightly affected by most parameters and is largely impaired when the sample number is 50 on AWA2. Softmax achieves the best performance when  $\sigma = 1$  and gradually becomes better when provided a larger sample number. Thus, our model is generally robust on diverse parameters.

## Conclusion

In this paper, we introduce a task-aligned meta generative model to mitigate the potential biases in visual-attribute correlation learning for zero-shot learning. We propose a task-wise distribution alignment method to enhance the current zero-shot learning and evaluate our model on four popular benchmark datasets to demonstrate TGMZ's strong capability to learn diverse task distributions under three ZSL settings. To better illustrate the learned features, we visualize the data space of the synthetic features, which is more discriminative than the state-of-the-art generative meta-ZSL method in most classes.

## References

- Al-Halah, Z.; and Stiefelagen, R. 2015. How to transfer? zero-shot object recognition via hierarchical transfer of semantic attributes. In *2015 IEEE Winter Conference on Applications of Computer Vision*, 837–843. IEEE.
- Changpinyo, S.; Chao, W.-L.; Gong, B.; and Sha, F. 2016. Synthesized classifiers for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5327–5336.
- Chen, C.; Dou, Q.; Chen, H.; and Heng, P.-A. 2018a. Semantic-aware generative adversarial nets for unsupervised domain adaptation in chest x-ray segmentation. In *International workshop on machine learning in medical imaging*, 143–151. Springer.
- Chen, L.; Zhang, H.; Xiao, J.; Liu, W.; and Chang, S.-F. 2018b. Zero-shot visual recognition using semantics-preserving adversarial embedding networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1043–1052.
- Day, O.; and Khoshgoftaar, T. M. 2017. A survey on heterogeneous transfer learning. *Journal of Big Data* 4(1): 29.
- Demertzis, K.; and Iliadis, L. 2020. GeoAI: A Model-Agnostic Meta-Ensemble Zero-Shot Learning Method for Hyperspectral Image Analysis and Classification. *Algorithms* 13(3): 61.
- Farhadi, A.; Endres, I.; Hoiem, D.; and Forsyth, D. 2009. Describing objects by their attributes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 1778–1785. IEEE.
- Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*.
- Firat, O.; Sankaran, B.; Al-Onaizan, Y.; Vural, F. T. Y.; and Cho, K. 2016. Zero-resource translation with multi-lingual neural machine translation. *arXiv preprint arXiv:1606.04164*.
- Gao, R.; Hou, X.; Qin, J.; Chen, J.; Liu, L.; Zhu, F.; Zhang, Z.; and Shao, L. 2020. Zero-VAE-GAN: Generating Unseen Features for Generalized and Transductive Zero-Shot Learning. *IEEE Transactions on Image Processing* 29: 3665–3680.
- Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.
- Guo, H.; Pasunuru, R.; and Bansal, M. 2020. Multi-Source Domain Adaptation for Text Classification via DistanceNet-Bandits. In *AAAI*, 7830–7838.
- Huang, H.; Wang, C.; Yu, P. S.; and Wang, C.-D. 2019. Generative dual adversarial network for generalized zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 801–810.
- Johnson, M.; Schuster, M.; Le, Q. V.; Krikun, M.; Wu, Y.; Chen, Z.; Thorat, N.; Viégas, F.; Wattenberg, M.; Corrado, G.; et al. 2017. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics* 5: 339–351.
- Kodirov, E.; Xiang, T.; and Gong, S. 2017. Semantic autoencoder for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3174–3183.
- Kumar, S.; Jat, S.; Saxena, K.; and Talukdar, P. 2019. Zero-shot word sense disambiguation using sense definition embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 5670–5681.
- Kumar Verma, V.; Arora, G.; Mishra, A.; and Rai, P. 2018. Generalized zero-shot learning via synthesized examples. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4281–4289.
- Lampert, C. H.; Nickisch, H.; and Harmeling, S. 2009. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 951–958. IEEE.
- Liu, L.; Zhou, T.; Long, G.; Jiang, J.; and Zhang, C. 2020a. Attribute Propagation Network for Graph Zero-Shot Learning. In *AAAI*, 4868–4875.
- Liu, S.; Long, M.; Wang, J.; and Jordan, M. I. 2018a. Generalized zero-shot learning with deep calibration network. In *Advances in Neural Information Processing Systems*, 2005–2015.
- Liu, Y.; Gao, Q.; Li, J.; Han, J.; and Shao, L. 2018b. Zero Shot Learning via Low-rank Embedded Semantic AutoEncoder. In *IJCAI*, 2490–2496.
- Liu, Z.; Yao, L.; Bai, L.; Wang, X.; and Wang, C. 2020b. Spectrum-guided adversarial disparity learning. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 114–124.
- Mishra, A.; Krishna Reddy, S.; Mittal, A.; and Murthy, H. A. 2018. A generative model for zero shot learning using conditional variational autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2188–2196.
- Nooralahzadeh, F.; Bekoulis, G.; Bjerva, J.; and Augenstein, I. 2020. Zero-shot cross-lingual transfer with meta learning. *arXiv preprint arXiv:2003.02739*.
- Pal, A.; and Balasubramanian, V. N. 2019. Zero-shot task transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2189–2198.
- Qin, Y.; Zhao, C.; Zhu, X.; Wang, Z.; Yu, Z.; Fu, T.; Zhou, F.; Shi, J.; and Lei, Z. 2019. Learning meta model for zero-and few-shot face anti-spoofing. *arXiv preprint arXiv:1904.12490*.
- Reed, S.; Akata, Z.; Lee, H.; and Schiele, B. 2016. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 49–58.



- Romera-Paredes, B.; and Torr, P. 2015. An embarrassingly simple approach to zero-shot learning. In *International Conference on Machine Learning*, 2152–2161.
- Sankaranarayanan, S.; Balaji, Y.; Castillo, C. D.; and Chellappa, R. 2018. Generate to adapt: Aligning domains using generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8503–8512.
- Shen, Y.; Liu, L.; Shen, F.; and Shao, L. 2018. Zero-shot sketch-image hashing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3598–3607.
- Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, 4077–4087.
- Soh, J. W.; Cho, S.; and Cho, N. I. 2020. Meta-Transfer Learning for Zero-Shot Super-Resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3516–3525.
- Sohn, K.; Lee, H.; and Yan, X. 2015. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, 3483–3491.
- Srivastava, S.; Labutov, I.; and Mitchell, T. 2018. Zero-shot learning of classifiers from natural language quantification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 306–316.
- Verma, V. K.; Brahma, D.; and Rai, P. 2020. Meta-Learning for Generalized Zero-Shot Learning. In *AAAI*, 6062–6069.
- Verma, V. K.; and Rai, P. 2017. A simple exponential family framework for zero-shot learning. In *Joint European conference on machine learning and knowledge discovery in databases*, 792–808. Springer.
- Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; et al. 2016. Matching networks for one shot learning. In *Advances in neural information processing systems*, 3630–3638.
- Wang, H.; Yang, W.; Lin, Z.; and Yu, Y. 2019a. TMDA: Task-Specific Multi-source Domain Adaptation via Clustering Embedded Adversarial Training. In *2019 IEEE International Conference on Data Mining (ICDM)*, 1372–1377. IEEE.
- Wang, Q.; Michau, G.; and Fink, O. 2020. Missing-class-robust domain adaptation by unilateral alignment. *IEEE Transactions on Industrial Electronics*.
- Wang, W.; Miao, C.; and Hao, S. 2017. Zero-shot human activity recognition via nonlinear compatibility based method. In *Proceedings of the International Conference on Web Intelligence*, 322–330.
- Wang, W.; Zheng, V. W.; Yu, H.; and Miao, C. 2019b. A survey of zero-shot learning: Settings, methods, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10(2): 1–37.
- Wang, X.; Yu, F.; Wang, R.; Darrell, T.; and Gonzalez, J. E. 2019c. TAFE-Net: Task-Aware Feature Embeddings for Low Shot Learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Welinder, P.; Branson, S.; Mita, T.; Wah, C.; Schroff, F.; Belongie, S.; and Perona, P. 2010. Caltech-UCSD birds 200 .
- Wilson, G.; and Cook, D. J. 2020. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)* 11(5): 1–46.
- Xian, Y.; Akata, Z.; Sharma, G.; Nguyen, Q.; Hein, M.; and Schiele, B. 2016. Latent embeddings for zero-shot classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 69–77.
- Xian, Y.; Lampert, C.; Schiele, B.; and Akata, Z. 2019. Zero-Shot Learning-A Comprehensive Evaluation of the Good, the Bad and the Ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41(9): 2251–2265.
- Xian, Y.; Lorenz, T.; Schiele, B.; and Akata, Z. 2018. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5542–5551.
- Zhang, H.; and Koniusz, P. 2018. Zero-shot kernel learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7670–7679.
- Zhang, L.; Xiang, T.; and Gong, S. 2017. Learning a deep embedding model for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021–2030.
- Zhang, Z.; and Saligrama, V. 2016. Zero-shot recognition via structured prediction. In *European conference on computer vision*, 533–548. Springer.
- Zhao, S.; Li, B.; Yue, X.; Gu, Y.; Xu, P.; Hu, R.; Chai, H.; and Keutzer, K. 2019. Multi-source domain adaptation for semantic segmentation. In *Advances in Neural Information Processing Systems*, 7287–7300.
- Zhu, Y.; Elhoseiny, M.; Liu, B.; Peng, X.; and Elgammal, A. 2018. A generative adversarial approach for zero-shot learning from noisy texts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1004–1013.
- Zhu, Y.; Xie, J.; Liu, B.; and Elgammal, A. 2019. Learning Feature-to-Feature Translator by Alternating Back-Propagation for Generative Zero-Shot Learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.