Attribute-Modulated Generative Meta Learning for Zero-shot Learning

Yun Li, Zhe Liu, Lina Yao, and Xiaojun Chang

Abstract—Zero-shot learning (ZSL) aims to transfer knowledge from seen classes to semantically related unseen classes, which are absent during training. The promising strategies for ZSL are to synthesize visual features of unseen classes conditioned on semantic side information and to incorporate meta-learning to eliminate the model’s inherent bias towards seen classes. While existing meta generative approaches pursue a common model shared across task distributions, we aim to construct a generative network adaptive to task characteristics. To this end, we propose an Attribute-Modulated generAtive meta-model for Zero-shot learning (AMAZ). Our model consists of an attribute-aware modulation network, an attribute-augmented generative network, and an attribute-weighted classifier. Given unseen classes, the modulation network adaptively modulates the generator by applying task-specific transformations so that the generative network can adapt to highly diverse tasks. The weighted classifier utilizes the data quality to enhance the training procedure, further improving the model performance. Our empirical evaluations on four widely-used benchmarks show that AMAZ outperforms state-of-the-art methods by 3.8% and 3.1% in ZSL and generalized ZSL settings, respectively, demonstrating the superiority of our method. Our experiments on a zero-shot image retrieval task show AMAZ’s ability to synthesize instances that portray real visual characteristics.

Index Terms—zero-shot learning, meta-learning, image retrieval.

I. INTRODUCTION

Object classification has undergone remarkable progress driven by the advances in deep learning. The underlying force ensuring the success is the availability of large amounts of carefully annotated image data. However, objects in the real world follow a long-tailed distribution [1], [2], i.e., a tremendous number of classes have few visual instances. Data insufficiency poses a bottleneck to the robustness of object classification methods. Targeting at overcoming this challenge, zero-shot learning has attracted plenty of interest recently [3]–[9].

Zero-shot learning (ZSL) aims to infer a classification model from seen classes, i.e., classes with labeled samples that present in the training process, to recognize unseen classes, i.e., classes that are absent from the training process. It generally leverages semantic side information to transfer knowledge from seen classes to unseen classes. Typical side information include human-defined attributes that portray visual characteristics [10]–[12], e.g., has tail, and word embeddings of text descriptions [13], [14].

A common strategy is to view ZSL as a visual-semantic embedding problem, which boils down to finding a projection that maps visual features and semantic features to the same latent space and performing nearest neighbor search in the space to predict labels [15]–[20]. However, this strategy suffers from the domain shift problem, due to distribution differences [21]. Some recent work uses generative methods to synthesize visual features conditioned on semantic side information and learn a conventional supervised classifier from generated samples to overcome the above issue [5]–[7], [22]–[24]. Given that generative models learned from seen classes exhibits inherent biases when generalizing to unseen classes, meta generative approaches for ZSL emerge as a new trend to mitigate the biases [8], [25]–[28]. Meta generative approaches incorporate meta-learning models, e.g., Model-Agnostic Meta-Learning (MAML) [29], into generative models. They divide seen classes into two disjoint sets (a support set and a query set) to mimic the ZSL setting and learn an optimal common generative model across seen and unseen classes.

Despite the effectiveness of meta generative approaches in ZSL, they still have limitations in learning characterized task distributions with diversities [30], e.g., in Computer Vision (CV) [31] and Natural Language Processing (NLP) [32]. First, the common model shared across tasks may be sub-optimal when applied to a specific task [33], [34]; it may result in a deteriorated model, which seeks a common solution while neglecting individual tasks’ characteristics. For example, given four species (i.e., sheep, horse, seal, and bobcat) from the Animals with Attributes (AwA) dataset [35]—which are highly

Fig. 1: Visualization of model modulation. $\theta$ denotes the learned model representation. $\theta^*_{a}, \theta^*_{b}, \theta^*_{c}$ and $\theta^*_{d}$ represent modulated models for four tasks. (a) Globally-shared model is sub-optimal for a single task. (b) Attribute-modulated model adapts $\theta$ according to task characteristics, e.g., attribute embedding, to fit highly diverse tasks, which obtains task-specific models.

Y. Li, Z. Liu, and L. Yao are with the School of Computer Science and Engineering, University of New South Wales, Sydney, NSW 2052, Australia (e-mail: yun.li5@student.unsw.edu.au; zheliu912@gmail.com; lina.yao@unsw.edu.au). X. Chang is with the Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: xiaojun.chang@uts.edu.au)
diverse—if we consider each class as a task, then, as shown in Figure 1 (a), the globally shared model representation $\theta$ learned by conventional meta generative methods would be far from being optimal for a single task. In few-shot learning, using samples to fine-tune the model can help relive the problem. But in ZSL, due to the lack of unseen data, the problem could be severe. Second, the learned model is not guaranteed to generate samples that can simulate or reflect characteristics of unseen classes due to the absence of real images of unseen classes during training. The synthetic data quality varies significantly across classes. The low-quality synthetic data may largely misguide and impair the training process of the final classifier.

In this paper, we generate features dynamically by proposing a novel Attribute-Modulated generAtive meta-model for Zero-shot learning (AMAZ). AMAZ can specialize a generalized model to adapt to diverse tasks, thus overcoming the first limitation. Specifically, we augment the meta generative adversarial network with an attribute-aware modulation network, which modulates layers within the generator according to task characteristics, as illustrated in Figure 1 (b). We propose an attribute discriminator for the modulation network to constrain its specialization direction. The specialization will be regularized towards the real data distribution. Moreover, we utilize the attribute discriminator to measure the synthetic data quality to modify the loss propagation of the weighted classifier. Thus, the weighted classifier can be robust to noisy low-quality data, which addresses the second limitation.

In summary, we make three-fold contributions:

- We propose an Attribute-Modulated generAtive meta-model for Zero-shot learning (AMAZ). AMAZ utilizes an attribute-aware modulation network to enhance the generative adversarial network and meta-learning. It combines the strengths of mitigating biases towards seen class and accommodating diverse tasks.
- We introduce data quality to provide complementary guidance for a weighted classifier. The weighted classifier improves performance on all datasets, especially in SUN (3.6%).
- We conduct extensive experiments in ZSL, Generalized-ZSL (GZSL), and zero-shot image retrieval tasks, where AMAZ consistently outperforms state-of-the-art algorithms on all four benchmarks, demonstrating our model’s superiority. Our ablation studies and further analysis also testify to our model’s robustness.

II. METHODOLOGY

A. Problem Definition

Let $D^S = \{(x, y, a) | x \in X^S, y \in Y^S, a \in A^S\}$ be the training data from seen classes, where $x \in X^S$ denotes the visual feature, $y \in Y^S$ denotes the class label of $x$, and $a \in A^S$ represents attributes (or any other kinds of semantic side information) of $y$. We define test data from unseen classes as $D^U = \{(x, y, a) | x \in X^U, y \in Y^U, a \in A^U\}$. Seen and unseen classes are disjoint, i.e., $Y^S \cap Y^U = \emptyset$. ZSL aims to learn a classifier $f_{ZSL} : x \in X^S$ that can classify objects from unseen classes. Generalized Zero-Shot Learning (GZSL) is more practical and challenging in that images from both seen and unseen classes may occur during the testing time: $f_{GZSL} : x \in X^U \cup X^S$. 

Fig. 2: Model Architecture of AMAZ. AMAZ is composed of a modulation network, a generative network, and an attribute-weighted classifier to fulfill the final classification. The modulation network takes attribute sets as input to produce parameters that are further used to modify the generator. The $\bigcirc$ denotes modulating operation. The $\bigcirc$ calculates cosine similarity.
In our model, we split $D^S$ into two disjoint sets $D_{sup}^S$ and $D_{qry}^S$ to function as support and query sets for meta-learning. We carry out episode-wise meta-training. In each episode, we sample task $T_i = (T_{sup}, T_{qry}) \sim p(T)$ from $D_{sup}^S$ and $D_{qry}^S$, where $p(T)$ denotes the task distribution over $D^S$. $T_{sup}$ and $T_{qry}$ are sampled in $N$-way $K$-shot setting, which means that each $T_i$ contains $N$ classes with $K$ labeled examples for each class. The number of $T_i$ in an episode is decided by a hyperparameter, i.e., batch size. We accumulate the gradients over all tasks in an episode for optimization.

The core of our proposed AMAZ (in Figure 2) is an attribute-modulated meta-generative network that synthesizes task-specific visual features based on the attributes. Our goal is to learn a Generator ($G$) $f_{\theta_g}(a,z) \rightarrow \hat{x}$, where $\theta_g$ denotes the model parameters of $G$; $(a,z)$ denotes the given attribute and random noise; $\hat{x}$ denotes the synthesized features. We use meta-learning and modulation network to specialize the generalized model representation based on the current task information to better handle diverse tasks. AMAZ consists of three components: attribute-aware modulation network, attribute-augmented generative network, and attribute-weighted classifier. The modulation network analyzes the attribute set of the current task to carry out task-specific model modulation. The generative network is modulated by the modulation network to fit the current tasks better and thus synthesize more accurate visual features. The weighted classifier measures the similarity of the synthesized features to enhance the training process of the final classifier. The training of AMAZ is based on episode-wise meta-learning, which enables the learned model parameters to be more generalized than conventional ZSL methods [8]. In the following sections, we will explain the model details and training procedures.

B. Attribute-aware Modulation Network

We use the attribute-aware modulation network to modulate the sub-optimal generator layer-wisely, then AMAZ can accommodate tasks with significant discrepancy. For each task in the training episode, $T_i$ consists of the objects from the current task. The attribute information can be denoted by $A_{T_i} = \{a_1, ..., a_N\}$, which contains attributes corresponding to $N$ classes in $T_i$. Then, we use an Attribute Projector (AP) to extract task embedding $e_i$ to represent the current task information:

$$e_i = f_{\theta_p}(A_{T_i})$$

where $\theta_p$ denotes the model parameter of AP; $e_i$ denotes the representation of task information of $T_i$; $A_{T_i}$ denotes the corresponding attribute set of $T_i$.

With the extracted task representation $e_i$, we use an Attribute Modulator (AM) to learn the modulation parameter $\{\{w, b\}\}$ that can adjust the layers in $G$:

$$(w_j, b_j) = h^j_{\theta_m}(e_i)$$

where $a_j$ denotes the intermediate result of the $j^{th}$ layer in $G$; $h^j_{\theta_m}$ denotes the $j^{th}$ AM to modulate $a_j$; $(w_j, b_j)$ denotes modulation parameters for $a_j$; $\hat{o}_j$ denotes the modulated output.

Eq. 3 shows the detailed modulation operation. Since the task information is same for a certain task $T_i$, we use the same input, i.e., $e_i$, for AMs. To produce different modulation parameters $\{\{w, b\}\}$, we use independent AMs for $G$, e.g., $j^{th}$ AM: $h^j_{\theta_m} \rightarrow \{(w_j, b_j)\}$ for $j^{th}$ layer.

We take $G$ with $k$ layers as an example and visualize the overall modulation flow in Figure 3. We layer-wisely modulate all the intermediate outputs in $G$ including the input $o_n = \{(z,a)\}$. The overlap of the $j^{th}$ layer and $j^{th}$ AM is performed layer by layer.

To enable AP and AM to learn correct task embeddings and modulate $G$ towards optimal directions, we design an Attribute Discriminator (AD):

$$a' = f_{\theta_{ad}}(f_{\theta_p}(a, z))$$

where $\theta_{ad}$ denotes the model parameters of AD; $a'$ is the reconstructed attribute.

We can view AD as a decoder, which attempts to reconstruct original input $a$. Therefore, we can compare $a'$ with $a$ to supervise the attribute-aware modulation network to improve the task information richness, which is consistent with the learning goals of AP and AM. Then, we can design the AD loss function $\mathcal{L}_{T_i}^{AD}$ to optimize AP and AM for task $T_i$:

$$\mathcal{L}_{T_i}^{AD} = \frac{1}{N*K} \sum_{n=1}^{N*K} \|a_n - a'_n\|_2^2$$

where $T_i$ contains $N*K$ samples; $a_n$ and $a'_n$ are the true attribute and the reconstructed attribute of $n^{th}$ sample in $T_i$.

Similarly, by comparing the similarity of $a$ and $a'$, we can infer the quality of the generated features. In other words, we use $a'$ to adjust the training of ZSL and GZSL classifiers to ease the influence of low-quality generation, which will be discussed in Section II-E.

C. Attribute-augmented Generative Network

The attribute-augmented generative network consists of three components: an attribute-aware Generator ($G$) modulated
by the modulation network to synthesize visual features: 
\( \hat{x} \leftarrow f_{\theta_g}(z, a) \); an auxiliary Classifier (C) to categorize the input samples: \( y' \leftarrow f_{\theta_c}(\hat{x}) \); a Discriminator (D) to distinguish real or fake visual features: \( \{0, 1\} \leftarrow f_{\theta_d}(\hat{x}, x) \).

For each task \( T_i \), following Eq. [3] and Figure [3] given a random noise \( z \in \mathbb{R} \) sampled from Gaussian distribution \( \mathcal{N}(0, \sigma) \), we can obtain the modulated synthetic visual features by \( \hat{x} = f_{\theta_g}(z, a) \). Then, we design the loss function of D for \( T_i \) as \( L_{D_{T_i}} \), which aims to optimize D to be able to distinguish fake features \( \hat{x} \sim f_{\theta_g}(a, z) \) as 0 and real features \( x \sim T_i \) as 1:

\[
L_{D_{T_i}} = \mathbb{E}_{a,z \sim T_i}[f_{\theta_d}(x, a)] - \mathbb{E}_{a,z}[f_{\theta_d}(f_{\theta_g}(a, z), a)]
\]

where \( \theta_d \) and \( \theta_g \) denote the parameters of G and D, respectively; \( z \sim \mathcal{N}(0, \sigma) \) denotes the random noise from normal distribution.

With the Eq. [6] we can obtain a reliable D to distinguish real and fake features. Then, we can optimize G to confuse D to synthesize more 'real' features. Besides from being real, we also need the classes of \( \hat{x} \) to be easy to predict, and G can collaborate with modulation network to better suit \( T_i \). Therefore, we apply an auxiliary classifier C to enhance the class information richness of \( \hat{x} \), and let \( L_{D_{T_i}}(a, a') \) be one of the learning goals of G. We design the loss function \( L_{GC_{T_i}} \) for generator as follows:

\[
L_{GC_{T_i}} = L_{D_{T_i}}(a, z) + L_{AD_{T_i}}(a, a') + L_{CLS_{T_i}}(y', y)
\]

\[
= -\mathbb{E}_{a,z \sim \mathcal{N}(0,\sigma)}[f_{\theta_d}(f_{\theta_g}(a, z), a)] + L_{AD_{T_i}}(a, a') + L_{CLS_{T_i}}(f_{\theta_d}(f_{\theta_g}(a, z)), y)
\]

where \( a' \) denotes the attribute vectors reconstructed by AD; \( y \) is the true class label; \( L_{CLS_{T_i}} \) is the classification loss measured using cross entropy. Note that we use \( L_{AD_{T_i}} \) in the optimization of AD to extract better attributes, but include it in \( L_{GC_{T_i}} \) to help GC generate semantic-rich samples.

With Eq. [7], we can construct a min-max loss function:

\[
\min_{\theta_p, \theta_M, \theta_{am}} \max_{\theta_d} L_{D_{T_i}} + L_{GC_{T_i}}
\]

where \( \theta_{am} = \{\theta_p, \theta_M, \theta_{ad}\} \) is the parameters of AP, AMs, and AD; \( \theta_{am} = \{\theta_m : j \in [1, k]\} \) is the set of AMs; \( \theta_d \) is the parameters of the D.

We optimize Eq. [8] in an adversarial manner through the following: 1) maximizing \( L_{D_{T_i}} \) to enable D to distinguish between real or generated samples; 2) optimizing \( \theta_{am} \) to minimize \( L_{GC_{T_i}} \) to enhance the quality of generated samples; 3) optimizing \( \theta_{gc} \), i.e., the generator and classifier, to minimize \( L_{GC_{T_i}} \) to fool the discriminator and assist attribute modulation network.

**D. Meta-training Procedure**

Following [29], we conduct episode-wise meta-training for our model in a model-agnostic manner (described in Algorithm [1]). In each iteration, we first sample tasks \( T_i = \{T_{sup}, T_{qry}\} \sim p(T) \), where \( T_{sup} \) and \( T_{qry} \) are sampled over \( D_{sup} \) and \( D_{qry} \) respectively (line 4). Considering a pre-split of training dataset could restrict the effectiveness of task sampling, in our experiment, we sample tasks from the whole training set—we only ensure the training and validation classes in each task are disjoint (line 3). After selecting classes for each task, we randomly sample images from these classes to construct support and query sets. Since \( T_{sup} \) and \( T_{qry} \) are disjoint to mimic seen and unseen classes in the ZSL setting, our AMAZ can learn to generate samples for unseen classes by transferring knowledge from seen classes.

Next, for each task, we modulate the generator according to the attribute set (lines 6 - 8). Then, \( T_{sup} \) is used for fast task adaptation (lines 10 - 15). We first optimize \( L_{D} \) and \( L_{AD} \) on support set to find the task-specific parameters for each task (lines 12 - 13), and then with updated \( D \) and \( AD \), we can optimize \( L_{GC} \) on support set (lines 15):

\[
\theta' \leftarrow \theta - \alpha \nabla_{\theta} L_{D_{T_{sup}}}(\theta)
\]

Note that D, AM, and GC are updated with different learning rate. Specifically, for D, the equation is \( \theta' \leftarrow \theta + \alpha \nabla_{\theta} L_{AD_{T_{sup}}}(\theta) \). We use \( \alpha \) for AM and GC to minimize \( L_{GC} \) and \( L_{AD} \), and use \( + \) for D to maximize \( L_{D} \).

Given the gradient from tasks in support set, we further use \( T_{qry} \) to update meta parameters, which are shared across all tasks, to obtain better generalized parameters (lines 17 - 19):

\[
\theta \leftarrow \theta - \beta \sum_{T_{qry}} \nabla_{\theta} L_{T_{qry}}(\theta')
\]

Also, all modules are updated with different learning rate, and \( L_{D} \) is optimized in direction opposite \( L_{AD} \) and \( L_{GC} \).

---

**Algorithm 1 AMAZ Training Procedure**

**Require:** \( p(T) \): task distribution, \( D_{S} \): training dataset

**Require:** \( \alpha_1, \alpha_2, \beta_1, \beta_2, \beta_3 \), batch size

1. Initialize \( \theta_d, \theta_g, \theta_{am} \)
2. **while** not DONE **do**
3. Split \( D_{S} \) into disjoint subsets \( D_{sup}^S \) and \( D_{qry}^S \)
4. Sample batches of tasks \( T_i = \{T_{sup}, T_{qry}\} \sim p(T) \) over \( D_{sup}^S \) and \( D_{qry}^S \) respectively
5. **for** all \( i \) **do**
6. Compute \( e_i \leftarrow f_{\theta_g}(A_{T_i}) \) with N classes in \( T_{sup}^i \)
7. **for** all \( j \) **do**
8. Generate \( (w_{ij}, b_{ij}) \leftarrow h_{\theta_m}(e_i) \) to modulate G
9. **end for**
10. Evaluate \( \nabla_{\theta_d} L_{D_{T_{sup}}}(\theta_d) \) w.r.t. samples in \( T_{sup}^i \)
11. Evaluate \( \nabla_{\theta_{am}} L_{AD_{T_{sup}}}(\theta_{am}) \) w.r.t. samples in \( T_{sup}^i \)
12. Update \( \theta'_d \leftarrow \theta_d + \alpha_1 \nabla_{\theta_d} L_{D_{T_{sup}}}(\theta_d) \)
13. Update \( \theta'_{am} \leftarrow \theta_{am} - \alpha_2 \nabla_{\theta_{am}} L_{AD_{T_{sup}}}(\theta_{am}) \)
14. Evaluate \( \nabla_{\theta_g} L_{GC_{T_{sup}}}(\theta_{gc}) \) w.r.t. samples in \( T_{sup}^i \)
15. Update \( \theta'_{gc} \leftarrow \theta_{gc} - \alpha_3 \nabla_{\theta_{gc}} L_{GC_{T_{sup}}}(\theta_{gc}) \)
16. **end for**
17. Update \( \theta_d \leftarrow \theta_d + \beta_1 \sum_{T_{qry}} \nabla_{\theta_d} L_{D_{T_{qry}}}(\theta_d') \)
18. Update \( \theta_{am} \leftarrow \theta_{am} - \beta_2 \sum_{T_{qry}} \nabla_{\theta_{am}} L_{AD_{T_{qry}}}(\theta_{am}) \)
19. Update \( \theta_{gc} \leftarrow \theta_{gc} - \beta_3 \sum_{T_{qry}} \nabla_{\theta_{gc}} L_{GC_{T_{qry}}}(\theta_{gc}) \)
20. **end while**
Different from MAML, during fast task adaptation, parameters are updated per batch-of-tasks instead of per task to enhance robustness [8].

E. Attribute-weighted Classifier

In this section, we first introduce weighted classifier based on the reconstructed attribute $a'$ from AD, and then introduce the inference process of ZSL/GZSL.

Weighted classifiers. ZSL/GZSL aims to train a classifier based on synthetic features to predict seen or unseen classes. Considering that the data quality of synthetic features differs from each other, we can adjust the loss weight based on the data quality to train the classifier to better fit the true data distribution and prevent fitting unreal features. We use cosine similarity between $a$ and $a'$ to measure the data quality:

$$Q(a') = \frac{a \cdot a'}{|a| \cdot |a'|}$$  \hspace{1cm} (11)

where $Q(a')$ denotes the data quality of $a'$; $\cdot$ denotes dot product; $|a|$ and $|a'|$ denote magnitude of $a$ and $a'$.

Then, we propose two weighted classifier based on Softmax (fully-connected layers followed by a Softmax layer) and Linear-SVM, respectively. For Softmax classifier which is trained in batch-wise, we can directly measure the instance-level data quality to adjust the gradient propagation to obtain more accurate classification loss:

$$\mathcal{L}_{\text{soft}} = \frac{1}{|A'|} \sum_{a' \in A'} Q(a', a) \cdot CE(\text{Softmax}(a'))\phi(a)$$  \hspace{1cm} (12)

where $a' \in A'$ denotes the generated features from current task; $a$ denotes the corresponding true attribute of $a'$; $|A'|$ denotes the sample number of $A'$; $CE$ denotes cross entropy loss; $\text{Softmax}(a') \rightarrow \hat{y}$ denotes the predicted label; $\phi(a) \rightarrow y$ denotes the corresponding true class label.

For Linear-SVM which is trained in dataset-wise, we propose a more efficient weighted loss function via class-level weights. We calculate class weights:

$$w_y = \frac{1}{|A_y|} \sum_{a' \in A_y} Q(a', a)$$  \hspace{1cm} (13)

where $A_y = \{a' : \phi(a') = y\}$ denotes the set of $a'$ whose true class label is $y$; $|A_y|$ denotes the sample number in $A_y$. Thus, we can calculate the weights for classes as the class weights for Linear-SVM to adjust the learning process.

Inference process. Given an unseen class, AMAZ first replicates its attribute $N$ times to construct attribute set, and then embeds the attribute set to produce attribute-aware parameters for modulation of the well-trained generator. Then, it applies the customized task-specific generator to synthesize visual features. By sampling $z$, an arbitrary number of visual features can be generated. With the generated visual features, a conventional supervised classifier, e.g., Linear-SVM and Softmax, can be trained to solve ZSL, i.e., the classifier only trained with synthesized features can be used to classify real features. In GZSL, samples for both seen and unseen classes are generated to train the final classifier. We use synthesized features instead of real data for seen classes to avoid bias.

### III. Experiments

#### A. Experiment Setup

Datasets: We conduct a comprehensive evaluation of our method in ZSL and GZSL settings on four widely used benchmark datasets: SUN [36], CUB [37], AWA1 [38], and AWA2 [39]. CUB and SUN are datasets of bird species and scenes, respectively; both are considered challenging since they are fine-grained and each class has limited data. AWA1 and AWA2 are coarse-grained datasets, where images come from highly diverse animals.

We adopt the commonly used 2048-dimensional CNN features extracted by ResNet101 [40] as visual features and use pre-defined attributes as semantic side information except for the CUB dataset. For CUB, we follow [8] to use CNN-RNN textual features as semantic information [41], which perform superior to the hand-hand-engineered attributes. Moreover, the datasets are divided into seen and unseen classes following the commonly used Proposed Split (PS) [39], and all the competitors use this split. Table I shows the statistics and splits of the datasets.

#### Implementation Details: We sample the support and query sets in 10-way-5-shot and 10-way-3-shot, respectively, and each batch has 10 tasks, i.e., each batch processes 800 images. The model is first optimized based on the 500 support images, and then the gradients are calculated by accumulating validation loss of the 300 images in query sets. We set the learning rates in Algorithm 1: $\alpha_1 = \alpha_2 = \alpha_3 = 1e-3$, $\beta_1 = 1e-3$, $\beta_2 = \beta_3 = 1e-5$. We use Adam optimizer to train the model. The epoch number is 25000 for SUN and CUB, and 15000 for AWA1 and AWA2. We set a larger number for SUN/CUB than for AWA1/2 because SUN/CUB has over 200 classes, while AWA1/2 only has 50 classes. We set the numbers to allow full training and report the final results. We set $\sigma$ to different values for training ($\sigma = 0.1$) and testing ($\sigma = 1$) to prevent the generator from being biased towards seen classes during testing. During testing, we consider the single class as a task and duplicate its attribute to modulate the generator first. Then, we use the modulated generator to synthesize visual features. We generate 100 samples for each unseen class in ZSL and 300 samples for both seen and unseen classes in GZSL. Network modules are implemented by the multi-layer perception. We adopt Dropout layer with parameter 0.5, Batch Normalize (BN) layer with parameter 0.8 and LeakyReLU (LeReLU) with parameter 0.5 as activation function in our network.

#### B. Zero-shot Learning

Table II shows the results of our experimental comparison with two groups (non-generative and generative) of methods.

| Datasets | Attribute dim | Image num | Seen/Unseen classes |
|-------|----------------|-----------|---------------------|
| AWA1  | 85             | 30455     | 40/10               |
| AWA2  | 85             | 37322     | 40/10               |
| CUB   | 1024           | 11788     | 150/50              |
| SUN   | 102            | 14340     | 645/72              |
image retrieval task on four datasets. We adopt two settings: 1) the guidance of attribute-weighted loss in training tasks; 2) the guidance of attribute-weighted loss in training classifier to denoise low-quality generated data.

![Image Retrieval Results](image-retrieval-results.png)

Figure 4 shows the results evaluated by average precision. AMAZ outperforms competitors by up to 5% in two settings, demonstrating that the attribute modulation network can enhance the generative quality. Besides, our model is more stable than competitors, reflected by minor error bars.

Component ablation study. Considering that ZSML is also a generative model trained by MAML, we take ZSML as our baseline. We compare ZSML and AMAZ adopting 4 different classifiers as ablations in ZSL. Our proposed attribute modulation network is effective on four benchmark datasets, demonstrated by the improvements by up to 2.8%, 1.7%, 4.6%, and 3.8% on SUN, CUB, AW A1, and AW A2, respectively. Besides, AMAZ exhibits more significant improvement on AW A1 and AW A2 than on SUN and CUB. It is reasonable because the objects are coarse-grained in AW A1 and AW A2 and have greater differences than the objects in SUN and CUB.

C. Generalized Zero-shot Learning

We follow [39] and calculate the average per-class Top-1 accuracy on seen (denoted by $S$) and unseen classes (denoted by $U$), and their harmonic mean (defined as $H = \frac{2UV}{U+V}$) to evaluate the performance in the GZSL setting. Table III shows the results comparing AMAZ with state-of-the-art methods. SVM is time and space consuming when classifying over 200 classes; thus we only report weighted-svm results for AW A2 and AW A1.

As shown in Table III, AMAZ surpasses all the other approaches and achieves 2.9%, 2.1%, 3.1%, and 1.9% improvements in harmonic mean on AW A2, AW A1, SUN, and CUB, respectively. Also, our model performs best on the unseen classes of AW A1, AW A2, and CUB, and second-best on SUN. This implies that our model can effectively infer visual features for unseen classes and eliminate bias towards seen classes. The improvements derive from three aspects: 1) the incorporation of meta-learning; 2) the use of attribute modulation network to make model better adapting to diverse tasks; 3) the guidance of attribute-weighted loss in training final classifiers to denoise low-quality generated data.

D. Zero-shot Image retrieval

Given the attributes of unseen classes, the zero-shot image retrieval problem aims to retrieve the most relevant images. We compare our model with three generative models, i.e., GAZSL [1], ZSML [8], and ABP [53], for the zero-shot image retrieval task on four datasets. We adopt two settings: retrieving the Top-5 and Top-10 images for each class from the whole dataset. Specifically, given the attributes, we utilize the generator to synthesize 100 features, calculate the average per-class Top-1 accuracy on seen (denoted by $S$) and unseen classes (denoted by $U$) to evaluate the performance in the GZSL setting. Table III shows the results comparing AMAZ with state-of-the-art methods. SVM is time and space consuming when classifying over 200 classes; thus we only report weighted-svm results for AW A2 and AW A1.

As shown in Table III, AMAZ surpasses all the other approaches and achieves 2.9%, 2.1%, 3.1%, and 1.9% improvements in harmonic mean on AW A2, AW A1, SUN, and CUB, respectively. Also, our model performs best on the unseen classes of AW A1, AW A2, and CUB, and second-best on SUN. This implies that our model can effectively infer visual features for unseen classes and eliminate bias towards seen classes. The improvements derive from three aspects: 1) the incorporation of meta-learning; 2) the use of attribute modulation network to make model better adapting to diverse tasks; 3) the guidance of attribute-weighted loss in training final classifiers to denoise low-quality generated data.
TABLE III: Overall comparison in GZSL. The performance is evaluated by average per-class Top-1 accuracy (%) on seen classes (S), unseen classes (U), and their harmonic mean (H). We embolden the best result and underline the second-best result on each dataset.

| Method            | AWA2   | AWA1   | SUN    | CUB    |
|-------------------|--------|--------|--------|--------|
|                   | U      | S      | H      | U      | S      | H      | U      | S      | H      |
| ESZSL [42]        | 5.9    | 77.8   | 11.0   | 11.0   | 27.9   | 15.8   | 12.6   | 63.8   | 21.0   |
| SYNC [49]         | 10.0   | 90.5   | 18.0   | 8.9    | 87.3   | 16.2   | 7.9    | 43.3   | 13.4   |
| DEM [50]          | 30.5   | 86.4   | 45.1   | 32.8   | 84.7   | 47.3   | 20.5   | 34.3   | 25.6   |
| Gaussian-Kernal [51] | 18.9   | 82.7   | 30.8   | 17.9   | 82.2   | 29.4   | 20.1   | 31.4   | 24.5   |
| TAFE-Net [16]     | 36.7   | 90.6   | 52.2   | 50.5   | 84.4   | 63.2   | 27.9   | 40.2   | 33.0   |
| PQZSL [52]        | 31.7   | 70.9   | 43.8   | 11.0   | 27.9   | 15.8   | 12.6   | 63.8   | 21.0   |
| SP-AEN [22]       | 23.0   | 90.9   | 31.7   | 27.0   | 67.9   | 38.6   | 22.2   | 38.3   | 26.5   |
| GAZSL [17]        | 35.4   | 86.9   | 50.3   | 29.6   | 84.2   | 43.8   | 22.1   | 39.3   | 28.3   |
| TVN [47]          | 51.7   | 74.8   | 61.1   | 50.5   | 67.8   | 57.9   | 49.0   | 26.0   | 40.5   |
| Zero-VAE-GAN [4]  | 32.9   | 78.7   | 46.4   | -      | -      | -      | -      | -      | -      |
| SELAR-GMP [7]     | 51.6   | 75.6   | 61.4   | 52.5   | 69.2   | 59.7   | 52.5   | 35.9   | 29.6   |
| ZSML [8]          | 60.1   | 69.2   | 64.3   | 64.4   | 63.6   | 64.1   | 64.0   | 65.3   | 64.1   |
| AMAZ weighted-soft (ours) | 35.4   | 86.9   | 50.3   | 29.6   | 84.2   | 43.8   | 22.1   | 39.3   | 28.3   |
| AMAZ weighted-svm (ours) | 56.0   | 74.6   | 64.0   | 57.6   | 75.5   | 63.6   | 42.0   | 35.1   | 38.3   |

Fig. 5: Visualization of image retrieval on CUB. Each row contains the five retrieved images given attributes of a specific class. The images circled by pink boxes are wrong results.

Warbler’. The possible reason is that ZSML, as a GAN-based model, is prone to mode collapse when CUB contains 200 fine-grained classes. In contrast, AMAZ, by introducing attribute discriminator and attribute loss, can generate highly discriminative features and achieve more inter-class diversity, thus avoiding mode collapse.

Overall, our model consistently outperforms competitors in different settings (as shown in Tables II and III), and different tasks (as shown in Figures 4 and 5), demonstrating the robustness and the superiority of AMAZ.

E. Modulator Operation Ablation Study

In this section, we exhibit ablation study on how the attribute-aware modulator modulate the generator. Given the intermediate results \( \{o_j\}_{j=0,1,...,k} \) of the generator and the attribute-aware parameters \( \{w_j, b_j\}_{j=0,1,...,k} \), the intuitive baseline design of the modulator is \( w_j o_j \). Based on the baseline, we further consider \( o_j \) as the base, \( b_j \) as the bias, activation function on \( w_j \) or \( b_j \) as activation, operator, e.g., +, that connecting different components as operator. The ZSL results of adopting different modulator on AWA2 are shown in Table IV. To avoid the influence of final classifier, we utilize

TABLE IV: Comparisons of the attribute-aware modulator operation. The performance is evaluated by average per-class Top-1 accuracy (%). w/o indicates without, and w the opposite.

| base | operator | activation | bias | Acc |
|------|----------|------------|------|-----|
| w/o  | w/o      | w/o        | w/o  | 10.8|
| w/o  | w/o      | Sigmoid    | w/o  | 22.9|
| w/o  | w/o      | Softmax    | w/o  | 22.5|
| w    | +        | Sigmoid    | w/o  | 72.0|
| w    | +        | Softmax    | w/o  | 71.2|
| w    | +        | -          | w/o  | 70.8|
| w    | +        | Softmax    | w/o  | 69.6|
| w    | +        | Sigmoid    | w    | 72.4|
SVM since its performance is more stable than Softmax. As shown in Table III, adding base and bias, using + as operator, and Sigmoid as activation are effective methods to improve the performance of the modulator.

F. Feature Visualization

We synthesize 500, 500, 100, and 50 features using ZSML and AMAZ for each unseen class in AWA2, AWA1, CUB, and SUN, respectively, and visualize the features with t-SNE. We chose the synthetic number to make sure the number of all synthetic features is around 5000. The generated features are visualized with t-SNE, and results are shown in Figure 6.

Legend of AWA1 is the same as AWA2. And for CUB and SUN, due to lack of space, legends are not plotted. In Figure 6(a), the features of ‘seal’, ‘dolphin’, and ‘walrus’ generated by ZSML highly overlap with each other, which makes sense since they are biologically similar. However, in Figure 6(b), these three unseen animals can be easily separated in our feature space, which validates our model’s superiority. Also, for AWA1 and CUB datasets, as shown in Figure 6(c) and (e), features generated by ZSML are highly overlapped for some classes, while features generated by AMAZ can be easily separated according to Figure 6(d) and (f). Besides, for SUN, from Figure 6(g) and (h), we can find that features generated by AMAZ are apparently easier to distinguish than features generated by ZSML.

Besides, as shown in Figure 6(e) and (f), the difference between the two visualization results is remarkably evident on CUB. The visualization of features generated by ZSML are highly scattered. The visualization difference is significant, while the performance gap is relatively trivial (1.7%). The underlying reason is that although we generated more discriminative samples, the generated samples may not reflect the real images of unseen classes due to domain shift. An example is that the attribute being ‘has tail’, the seen class being monkey, and the unseen class being horse; although our generated sample can be more discriminative via better reflecting ‘has tail’, the generated features for horse may still be insufficient for training since monkey and horse have tails of different appearance.

G. Hyper-parameter Analysis

Fig. 7: Hyper-parameter analysis in ZSL. Shadow alongside the curves represents standard deviation.

We conduct ablation studies to investigate how the number of synthetic samples and σ of noise z influence ZSL and GZSL. Experiments are done by our weighted-soft version classifier.

Zero-shot Learning: The results in Figure 7(a) and (b) show that AMAZ achieves robust performance, which changes slightly on the four datasets when parameter values increase.
AMAZ achieves satisfactory results when $\sigma = 1$ and sample number reaches 100. The performance on AWA1 and AWA2 is more impacted by parameters, reflected by larger standard deviations in per-class accuracy. The reason is that seen and unseen classes are more different in AWA1 and AWA2, thus introducing a greater bias.

**Generalized Zero-Shot Learning**: We utilize real data for seen classes and synthetic features for unseen classes to train the classifier. The curves of the harmonic mean (H-SUN and H-CUB) in Figure 8 show that comparing with ZSL, the model performance is more stable when varying synthetic numbers and $\sigma$. The reason is that classifier needs to predict labels for more classes on CUB and SUN in GZSL (50 vs. 200, and 72 vs. 717); thus only significant improvement or drop of the model performance can result in fluctuation of H.

IV. RELATED WORK

A. Zero-shot Learning

A common strategy views zero-shot learning as an embedding problem of visual or semantic features. For example, Ye et al. [18] design a progressive ensemble network for learning a mapping function from the same extracted features to different label representations. Bendre et al. [54] propose multi-modal caritational autoencoder based on a multi-modal loss to correlate modalities and global-local semantic knowledge for ZSL. Han et al. [55] utilize the mutual information to learn the redundancy-free visual embedding for better discrimination of features. Imrattanatra et al. [56] propose an embedding model based on a knowledge graph. Such methods resort to learning a projection from visual space to semantic space [18], [44], [45], [50]-[58] or the reverse [50], [59]. Then, ZSL can be accomplished by ranking similarity or compatibility in the shared space. Embedding can be learned based on the features extracted utilizing pre-trained backbones (e.g., ResNet101) or in an end-to-end manner [60]. Zhang et al. [50] propose a multi-modality fusion method that enables end-to-end learning of semantic descriptors. Moreover, Guo et al. [61] propose a one-step recognition framework to perform recognition in the original feature space and thus avoid information loss of the intermediate transformation.

In the more challenging GZSL setting, where only instances from seen classes are provided, the embedding methods are more prone to suffer from data imbalance in recognizing data from both seen and unseen classes. Conventional embedding methods cannot address such problems well [39].

To address the data imbalance issue, several others efforts [5], [6], [7], [47], [62]-[65] explore generative methods for ZSL. Felix et al. [73] use a generative adversarial network to synthesize features constrained by multi-modal cycle-consistent semantic compatibility. Variational autoencoder [66]-[68] is also adopted to avoid mode collapse caused by the structure of GAN. The f-CLSWGAN [46] synthesizes the unseen instances according to the semantic descriptions. Zero-VAE-GAN [6] combines two generative models, variational autoencoder (VAE) and generative adversarial networks (GAN), to improve the model’s performance and robustness. Xu et al. [62] adopt two couple Wassertein GANs to generate semantic-related multi-modal features for further image retrieval. By generating samples for unseen classes, ZSL is converted to supervised classification, and conventional classification methods can be applied. In our model, we propose an attribute-weighted loss to enhance both deep learning and traditional machine learning classifiers.

Recently, meta-model is proposed to further eliminate the bias towards seen classes in ZSL [8], [25]-[28], [69]. For example, Yu et al. [70] first introduce an episode-based training framework for ZSL, which can progressively accumulate ensemble experiences based on the mimetic unseen classes and thus generalize the semantic prototypes for real unseen classes. TAFE-NET [16] uses a meta learner for task-aware feature embedding. ZSML [8] introduce and meta-learning and generative network in ZSL. Liu et al. [69] propose to utilize a task-wise attribute alignment network to mitigate the potential biased meta-learning. They all rely on MAML [29] and generative model. Although meta-models have achieved great success, they seek a common solution to be shared across tasks and thus fail to accommodate diverse or new tasks. To address this limitation, our model introduces a modulation network and promises attribute-aware meta-learning.

Considering that generative methods cannot guarantee the quality of generated data for unseen classes due to the absence of real images, transductive methods are proposed to address the limitation [6], [71], [71], [72]. Differing from ZSL, transductive methods assume that unlabeled images from unseen classes may occur during training and use the unlabeled images as auxiliary information. Since this assumption is not strict as ZSL, we adopt the traditional inductive ZSL yet improve its generating quality by employing the attribute-aware modulation network. The attribute modulator modifies the generator towards attribute-awareness and attribute-richness.

B. Feature Modulation

Our AMAZ is also related to feature modulation, which explores the modulation of fully connected networks or convolutional networks. Some research [73]-[75] introduces conditional batch normalization to modulate a target neural network’s visual processing based on linguistic input. For example, Yu et al. [76] directly generate the classifiers based on the class descriptions and semantic information of target unseen classes. Li et al. [6] use the area under score curve as weights to adapt tasks of ZSL and thus learn characterized semantic concepts. Our attribute-aware modulation can be viewed as an episode-wise feature modulation conditioned on attributes. But inspired by attention mechanism [77], our model applies gate value, e.g., Sigmoid, to generated parameters before modulation. The gate value helps emphasize more attribute-related features and thus removes redundancy from synthetic instances. Compared with previous work [73]-[75], our model is more suitable for zero-shot learning, as it modulates the generator to reflect the semantic characteristics.

C. Model-Agnostic Meta-Learning

Model-Agnostic Meta-Learning (MAML) [29] is a meta-learning based optimization framework. It aims to find a initial model, which can be fast adapted to other tasks with
few samples. The authors achieve the goal by designing a two-step optimization procedure: meta-training and meta-validation. During meta-training, MAML optimizes the model towards different directions to fit different tasks. Then, during meta-validation, MAML aggregates the gradients from all optimization directions in meta-training and minimizes the overall validation loss based on validation tasks. Our model follows the same training procedure of MAML except that we update parameters per batch instead of per task to improve robustness.

V. CONCLUSION

In this paper, we present an attribute-modulated generative meta-model (AMAZ) to synthesize visual features for unseen classes for ZSL. AMAZ incorporates an attribute-aware modulation network to modify the generator according to task characteristics learned from attributes. It introduces an attribute discriminator to guide the direction of modulation. Then the customized generator can be tuned for each task and adapt to diverse tasks, including new tasks. In addition, AMAZ is trained in an episode-wise meta manner to mitigate the inherent bias caused by the absence of unseen data during training. We further improve the AMAZ by an attribute-weighted classifier, which can denoise low-quality synthetic data. Extensive experiments on four widely-used benchmarks show that our model exceeds state-of-the-art methods in both ZSL and GZSL settings. The qualitative and quantitative experiments in zero-shot image retrieval also show that AMAZ generates more discriminative features. In the future, we will extend the model to address the insufficiency of labeled data for more complex retrieval tasks, such as fine-grained image retrieval and cross-media retrieval.

REFERENCES

[1] Y. Zhu, M. Elhoseiny, B. Liu, X. Peng, and A. Elgammal, “A generative adversarial approach for zero-shot learning from noisy texts,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 1004–1013.

[2] L. Yang, C. Kong, X. Chang, S. Zhao, Y. Cao, and S. Zhang, “Correlation filters with adaptive convolution response fusion for object tracking,” Knowl. Based Syst., vol. 228, p. 107314, 2021.

[3] W. Wang, Y. Pu, V. K. Verma, K. Fan, Y. Zhang, C. Chen, P. Rai, and L. Carin, “Zero-shot learning via class-conditioned deep generative models,” in Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[4] V. K. Verma and P. Rai, “A simple exponential family framework for zero-shot learning,” in Joint European conference on machine learning and knowledge discovery in databases. Springer, 2017, pp. 792–808.

[5] P. Zhu, H. Wang, and V. Saligrama, “Generalized zero-shot recognition based on visually semantic embedding,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2995–3003.

[6] R. Gao, X. Hou, J. Qin, J. Chen, L. Liu, F. Zhu, Z. Zhang, and L. Shao, “Zero-vae-gan: Generating unseen features for generalized and transductive zero-shot learning,” IEEE Transactions on Image Processing, vol. 29, pp. 3665–3680, 2020.

[7] S. Yang, K. Wang, L. Herranz, and J. van de Weijer, “Simple and effective localized attribute representations for zero-shot learning,” arXiv, pp. arXiv–2006.2020, 2020.

[8] V. K. Verma, D. Brahma, and P. Rai, “Meta-learning for generalized zero-shot learning,” in AAAI, 2020, pp. 6002–6009.

[9] Z. Li, L. Yao, X. Chang, K. Zhan, J. Sun, and H. Zhang, “Zero-shot event detection via event-adaptive concept relevance mining,” Pattern Recognition, vol. 88, pp. 595–603, 2019.

[10] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, “Describing objects by their attributes,” in 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2009, pp. 1778–1785.

[11] A. Farhadi, I. Endres, and D. Hoiem, “Attribute-centric recognition for cross-category generalization,” in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, 2010, pp. 2352–2359.

[12] M. Luo, X. Chang, and C. Gong, “Reliable shot identification for complex event detection via visual-semantic embedding,” Comput. Vis. Image Underst., vol. 213, p. 103300, 2021.

[13] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele, “Evaluation of output embeddings for fine-grained image classification,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 2927–2936.

[14] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

[15] S. Liu, M. Long, J. Wang, and M. I. Jordan, “Generalized zero-shot learning with deep calibration network,” in Advances in Neural Information Processing Systems, 2018, pp. 2005–2015.

[16] X. Wang, F. Yu, R. Wang, T. Darrell, and J. E. Gonzalez, “Tafe-net: Task-aware feature embeddings for low shot learning,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 1831–1840.

[17] L. Liu, T. Zhou, G. Long, J. Jiang, and C. Zhang, “Attribute propagation network for graph zero-shot learning,” in AAAI, 2020, pp. 4868–4875.

[18] M. Ye and Y. Guo, “Progressive ensemble networks for zero-shot recognition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 11728–11736.

[19] Y. Hu, G. Wen, A. Chapman, P. Yang, M. Luo, Y. Xu, D. Dai, and W. Hall, “Semantic graph-enhanced visual network for zero-shot learning,” arXiv preprint arXiv:2006.04648, 2020.

[20] P. Ren, Y. Xiao, X. Chang, P. Huang, Z. Li, X. Chen, and X. Wang, “A comprehensive survey of neural architecture search: Challenges and solutions,” ACM Comput. Surv., vol. 54, no. 4, pp. 76:1–76:34, 2021.

[21] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong, “Transductive multi-view zero-shot learning,” IEEE transactions on pattern analysis and machine intelligence, vol. 37, no. 11, pp. 2332–2345, 2015.

[22] L. Chen, H. Zhang, J. Xiao, W. Liu, and S.-F. Chang, “Zero-shot visual recognition using semantics-preserving adversarial embedding networks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1043–1052.

[23] R. Felix, V. B. Kumar, I. Reid, and G. Carneiro, “Multi-modal cycle-consistent generalized zero-shot learning,” in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 21–37.

[24] Y. Xiao, W. Lei, L. Lu, X. Chang, X. Zheng, and X. Chen, “CS-GAN: cross-structure generative adversarial networks for chinese calligraphy translation,” Knowl. Based Syst., vol. 229, p. 107334, 2021.

[25] A. Pal and V. N. Balasubramanian, “Zero-shot task transfer,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2019, pp. 2189–2198.

[26] J. W. Soh, S. Cho, and N. I. Cho, “Meta-transfer learning for zero-shot super-resolution,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 3516–3525.

[27] K. Demertzis and L. Iliadis, “Geoai: A model-agnostic meta-ensemble zero-shot learning method for hyperspectral image analysis and classification,” Algorithms, vol. 13, no. 3, p. 61, 2020.

[28] F. Nooralahzadeh, G. Bekoulis, J. Bjerva, and I. Augenstein, “Zero-shot cross-lingual transfer with meta learning,” arXiv preprint arXiv:2003.02739, 2020.

[29] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in ICML, 2017.

[30] S. Sinha, H. Bharadhwaj, A. Goyal, H. Larochelle, A. Garg, and F. Shkurti, “Dibs: Diversity inducing information bottleneck in model ensembles,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 11, 2021, pp. 9666–9674.

[31] E. David, S. Madec, P. Sadeghi-Tehrani, H. Aasen, B. Zheng, S. Liu, N. Kirchgessner, G. Ishikawa, K. Nagasawa, M. A. Badhon et al., “Global wheat head detection (gwhd) dataset: a large and diverse dataset of high-resolution rgb-labelled images to develop and benchmark wheat head detection methods,” Plant Phenomics, vol. 2020, 2020.

[32] E. Reuver, A. Fokkens, and S. Verbeeck, “No ppl task should be an island: Multi-disciplinarity for diversity in news recommender systems,” in Proceedings of the EACL Hackshop on News Media Content Analysis and Automated Report Generation, 2021, pp. 45–55.
Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, “Feature generating F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, E. Kodirov, T. Xiang, and S. Gong, “Semantic autoencoder for zero-shot learning,” in 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 951–958.

Y. Xian, C. Lampert, B. Schiele, and Z. Akata, “Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 9, pp. 2251–2269, 2015.

K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

S. Reed, Z. Akata, H. Lee, and B. Schiele, “Learning deep representations of fine-grained visual descriptions,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 49–58.

B. Romero-Paredes and P. Torr, “An embarrassingly simple approach to zero-shot learning,” in International Conference on Machine Learning, 2015, pp. 2152–2161.

Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele, “Latent embeddings for zero-shot classification,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 69–77.

E. Kooij, T. Xiang, and S. Gong, “Semantic autoencoder for zero-shot learning,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3174–3183.

F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, “Learning to compare: Relation network for few-shot learning,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1199–1208.

Y. Xian, T. Lorentz, B. Schiele, and Z. Akata, “Feature generating networks for zero-shot learning,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 5542–5551.

H. Zhang, Y. Long, Y. Guan, and L. Shao, “Triple verification network for generalized zero-shot learning,” IEEE Transactions on Image Processing, vol. 26, no. 1, pp. 506–517, 2019.

X. Chen, J. Li, X. Lan, and N. Zheng, “Generalized zero-shot learning via multi-modal aggregated posterior aligning neural network,” IEEE Transactions on Multimedia, 2020.

S. Changpipinyo, W.-L. Chao, B. Gong, and F. Sha, “Synthesized classifiers for zero-shot learning,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 5327–5336.

L. Zhang, T. Xiang, and S. Gong, “Learning a deep embedding model for zero-shot learning,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2021–2030.

H. Zhang and P. Koniusz, “Zero-shot kernel learning,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7658–7667.

J. Li, X. Lan, X. Liu, L. Wang, and N. Zheng, “Compressing unknown images with product quantizer for efficient zero-shot classification,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5463–5472.

Y. Zhu, J. Xie, B. Liu, and A. Elgammal, “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), Oct 2019.

N. Bendre, K. Desai, and P. Najafirad, “Generalized zero-shot learning using multimodal variational auto-encoder with semantic concepts,” in 2021 IEEE International Conference on Image Processing (ICIP). IEEE, 2021, pp. 1284–1288.

Z. Han, Z. Fu, and J. Yang, “Learning the redundancy-free features for generalized zero-shot object recognition,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 12865–12874.

W. Innaratthanan, M. P. Kato, and M. Yoshikawa, “Identifying entity properties from text with zero-shot learning,” in Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 195–204.

X. Lu, L. Liu, L. Nie, X. Chang, and H. Zhang, “Semantic-driven interpretation: Deep multi-modal hashing for large-scale multimedia retrieval,” IEEE Trans. Multimedia, vol. 23, no. 4, pp. 4541–4554, 2021.

J. Li, M. Jing, L. Zhu, Z. Ding, K. Lu, and Y. Yang, “Learning modality-invariant latent representations for generalized zero-shot learning,” in Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 1348–1356.

Y. Shigeto, I. Suzuki, K. Hara, M. Shimbo, and Y. Matsumoto, “Ridge regression, hubness, and zero-shot learning,” in Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, 2015, pp. 135–151.

W. Guan, X. Song, T. Gan, J. Lin, X. Chang, and L. Nie, “Cooperation learning from multiple social networks: Consistent and complementary perspectives,” IEEE Trans. Cybern., vol. 51, no. 9, pp. 4501–4514, 2021.

Y. Guo, G. Ding, J. Han, and Y. Gao, “Zero-shot learning with transferred samples,” IEEE Transactions on Image Processing, vol. 26, no. 7, pp. 3277–3290, 2017.

X. Xu, K. Lin, H. Lu, L. Gao, and H. T. Shen, “Correlated features synthesis and alignment for zero-shot cross-modal retrieval,” in Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 1419–1428.

G. Yang, J. Liu, and X. Li, “Imagination based sample construction for zero-shot learning,” in The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018, pp. 941–944.

O. Gune, B. Banerjee, S. Chaudhuri, and F. Cuzzolin, “Generalized zero-shot learning using generated proxy unseen samples and entropy separation,” in Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 4262–4270.

C. Yan, X. Chang, Z. Li, W. Guan, Z. Ge, L. Zhu, and Q. Zheng, “Zeronas: Differentiable generative adversarial networks search for zero-shot learning,” IEEE Trans. Pattern Anal. Mach. Intell., 2021.

A. Mishra, S. Krishna Reddy, A. Mittal, and H. A. Murphy, “A generative model for zero-shot learning using conventional variational auto-encoders,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 2188–2196.

E. Schonfeld, S. Ebrahimi, S. Sinha, T. Darrell, and Z. Akata, “Generalized zero-and few-shot learning via aligned variational autoencoders,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 8247–8255.

P. Ma and X. Hu, “A variational autoencoder with deep embedding model for generalized zero-shot learning,” in AAAI, 2020, pp. 11 733–11 740.

Z. Liu, Y. Li, L. Yao, X. Wang, and G. Long, “Task aligned generative meta-learning for zero-shot learning,” in Proceedings of The Thirty-Fifth AAAI Conference on Artificial Intelligence, 2021.

Y. Yu, Z. Ji, J. Han, and Z. Zhang, “Episode-based prototype generating network for zero-shot learning,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 14035–14044.

Z. Wan, D. Chen, Y. Li, X. Yan, J. Zhang, Y. Yu, and J. Liao, “Transductive zero-shot learning with visual structure constraint,” in Advances in Neural Information Processing Systems, 2019, pp. 9972–9982.

S. Rahman, S. Khan, and N. Barnes, “Transductive learning for zero-shot object detection,” in Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 6082–6091.

K. Zhang, J. Chen, B. Liu, and Q. Liu, “Deep object co-segmentation via spatial-semantic network modulation,” in AAAI, 2020, pp. 12 813–12 820.

H. De Vries, F. Strub, J. Mary, H. Larochelle, O. Pietquin, and A. C. Courville, “Modulating early visual processing by language,” in Advances in Neural Information Processing Systems, 2017, pp. 6594–6604.

E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, “Film: Visual reasoning with a general conditioning layer,” arXiv preprint arXiv:1709.07871, 2017.

Y. Yu, B. Li, Z. Ji, J. Han, and Z. Zhang, “Knowledge distillation classifier generation network for zero-shot learning,” IEEE Transactions on Neural Networks and Learning Systems, 2021.

F. Zhang, I. Goodfellow, and A. Ozdena, “Self-attention generative adversarial networks,” in International Conference on Machine Learning, 2019, pp. 7354–7363.