GSMFlow: Generation Shifts Mitigating Flow for Generalized Zero-Shot Learning
Zhi Chen, Yadan Luo, Sen Wang, Jingjing Li, and Zi Huang, Member, IEEE

Abstract—Generalized Zero-Shot Learning (GZSL) aims to recognize images not only for seen classes but also for unseen ones by transferring semantic-visual relationships from the seen to the unseen classes. It is an intuitive solution to take the advantage of generative models to hallucinate realistic unseen samples based on the knowledge learned from the seen classes. However, due to the generation shifts, the synthesized samples by most existing methods may drift from the real distribution of the unseen data. To address this issue, we propose a novel flow-based generative framework that consists of multiple conditional affine coupling layers for learning unseen data generation. Specifically, we investigate and address three essential problems that trigger the generation shifts, i.e., semantic inconsistency, variance collapse, and structure disorder. First, to mitigate the drift of the semantic information in the generated samples, we proactively embed the semantic information into the transformation in each conditional affine coupling layer. Second, to promote the intrinsic feature variance of the unseen classes, we introduce a boundary sample mining strategy with entropy maximization to discover ambiguous visual variants of semantic prototypes and hereby calibrate the decision boundary of the classifiers. Third, a relative positioning strategy is proposed to revise the attribute embeddings, guiding which to fully preserve the inter-class geometric structure and further avoid structure disorder in the semantic space. Extensive experimental results on four GZSL benchmark datasets demonstrate that GSMFlow achieves the state-of-the-art performance on GZSL.

Index Terms—Deep learning, generative flow, zero-shot learning.

I. INTRODUCTION

TRADITIONAL visual recognition systems have progressed substantially with the help of a massive amount of labeled data and deep learning techniques. However, it is challenging for these recognition systems to generalize to unseen classes that are unknown during training. Also, the need for large-scale labeled data has been a curse when training a deep recognition system. Generally, one may curate a certain amount of data for every individual class in order to train a canonical classification model, while the desire for training data samples is drasticaly amplified for deep learning methods. Data labeling is time-consuming and expensive, even if the raw data is usually plentiful. Moreover, it is unrealistic to require labeled data for every class, leading to extra attention from researchers drawn to Zero-Shot Learning (ZSL) as a solution [1]–[5]. By incorporating the class-level side information, e.g., semantic attributes, ZSL transfers the semantic-visual relationships from seen classes to unseen classes without the need of visual samples for unseen classes. While conventional ZSL aims to recognize only unseen classes, it is infeasible to assume that we will only come across samples from unseen classes. Hence, in this paper, we consider a more realistic and challenging task, Generalized Zero-Shot Learning (GZSL), to classify over both seen and unseen classes [6]–[9].

GZSL can be roughly categorized into embedding-based methods [1]–[3], [10], [11] and generative methods [7], [8], [12]–[15]. Embedding-based methods usually project the semantic and visual features into the same space, where the compatibility scores between the visual features and all classes are computed for predictions. In contrast, generative methods [7], [8], [16]–[19], as shown in Fig. 1, convert the GZSL problem into supervised classification tasks, where synthesized visual features for unseen classes are generated and a supervised classifier will be trained on both seen visual features and the synthesized unseen visual features.

In spite of the advances achieved by the generative paradigm, the synthesized visual features may not be guaranteed to accurately represent the real distribution of unseen classes, given that the unseen features are not exposed during training. Due to the gap between the synthesized and the real distributions of unseen features, the trained classifier tends to misclassify the unseen
A novel GSMFlow framework is proposed for GZSL, where a small amount of noise is further added to bring more visual variants for each class, which will calibrate the decision boundary for suitin the real unseen samples at the test time. Moreover, to alleviate the semantic structure disorder, the relative positioning mechanism is adopted to correct the attribute representations by preserving the global geometrical relationship to the specific semantic anchors. Specifically, the visual features $x$ of a real sample are first extracted from a backbone network, e.g., ResNet101. In the training stage, the boundary samples that represent more extreme visual variants are learned from the visual samples in the original dataset, in addition to which a small amount of noise is further added to bring more diversity. The resulted diverse and hard visual samples are fed into a flow-based generative framework. The training process is conditioned on the manipulated global semantic vector to infer the latent factors drawn from a prior distribution. The learned generative flow has its inverse transformation as the generative network. Similarly, a global semantic vector is progressively injected into each of the inverse coupling layers to generate unseen visual features from latent factors. Eventually, a unified classifier is trained with the real seen features and the synthesized unseen features, aiming to accurately recognize both seen and unseen classes at the test time. To sum up, the contributions of our work are listed as follows:

- A novel GSMFlow framework is proposed for GZSL, which proactively incorporates the class-level semantic information into both the forward and the inverse transformations of the conditional generative flow. It encourages the synthesized samples to be more coherent with the respective semantic information.
- We propose a boundary sample mining strategy to learn the visual representation boundaries for each semantic vector. Moreover, a visual perturbation strategy is proposed to imitate the intrinsic variance of unseen classes in synthesized samples. By learning more boundary samples and injecting perturbation noise, we diversify the training samples to enrich the original feature space. As the generative flow is exposed to more diverse virtual samples, the synthesized samples of unseen classes can thus capture more visual potentials.
- To preserve the geometric relationship between different semantic vectors in the semantic space, we choose different semantic anchors to revise the representation of the attributes.
- Comprehensive experiments and in-depth analysis on four GZSL benchmark datasets demonstrate the state-of-the-art performance by the proposed GSMFlow framework in the GZSL tasks.

A preliminary version of GSMFlow is presented in [20], which is accepted by ACM Multimedia 2021. We have made extensive revision and expansion in this paper. To address the variance collapse issue, in the preliminary version, we proposed a visual perturbation strategy to imitate the intrinsic variance of unseen classes in synthesized samples. In this paper, we further propose a boundary sample mining technique with entropy maximization that comprehensively enriches the visual space to prevent variance collapse. Moreover, more detailed discussions and analysis are provided in this paper.

The rest of the paper is organised as follows. We briefly review related work in Section II. GSMFlow is presented in Section III, followed by the experiments and the in-depth analysis in Section IV. Lastly, Section V concludes the paper.

II. RELATED WORK

A. Traditional Zero-Shot Learning

Traditional solutions towards zero-shot learning are mostly the embedding-based methods. The pioneering method ALE [1]
proposes to employ embedding functions to measure the compatibility scores between a semantic embedding and a data sample. SJE [3] extends ALE by using structured SVM [21] and takes advantage of the structured outputs. DeViSE [22] constructs a deep visual semantic embedding model to map 4096-dimensional visual features from AlexNet [23] to 500 or 1000-dimensional skip-gram semantic embeddings. EZSL [24] theoretically gives the risk bound of the generalization error and connects zero-shot learning with the domain adaptation problem. More recently, SAE [25] develops a cycle embedding approach with an autoencoder to reconstruct the learned semantic embeddings into a visual space. Later, the cycle architecture is also investigated in generative methods [26]. However, these early methods have not achieved satisfactory results on zero-shot learning. Particularly, when applying on the GZSL task, the unseen class performance is even worse. Recently, thanks to the advances of generative models, with which the visual samples of unseen classes can be augmented to train a supervised classifier on both seen and unseen classes. Zero-shot learning can then be converted into supervised classification tasks.

B. Generative GZSL

A number of generative methods have been applied for GZSL, e.g., Generative Adversarial Nets (GANs) [27], Variational Autoencoders (VAEs) [28], and Alternating Back-Propagation algorithms (ABPs) [29]. f-CLSWGAN [8] presents a WGAN-based [30] approach to synthesize unseen visual features based on semantic information. CADA-VAE [7] proposes to stack two VAEs, each for one modality, and aligns the latent spaces. The latent space, thus, can enable information sharing among different modality sources. LisGAN [12] is inspired by the multi-view property of images and improves f-CLSWGAN by encouraging the generated samples to approximate at least one visually representative view of samples. CANZSL [26] considers the cycle-consistency principle of image generation and proposes a cycle architecture by translating synthesized visual features into semantic information. ABP-ZSL [31] adopts the rarely studied generative models ABPs to generate visual features for unseen classes. GDAN [6] incorporates a flexible metric in the model’s discriminator to measure the similarity of features from different modalities.

The bidirectional conditional generative models enforce cycle-consistent generation and allow the generated images to truthfully reflect the conditional information [26], [32]. Instead of encouraging cycle consistency through adding additional reverse networks, generative flows are bidirectional and cycle-consistent in nature. The generative flows are designed to infer and generate within the same network. Also, the generative flows are lightweight comparing to other methods as no auxiliary networks are needed, e.g., discriminator for GANs, variational encoder for VAEs. Some conditional generative flows [33], [34] are proposed to learn image generation from class-level semantic information. IZF [18] adopts the invertible flow model [33] for GZSL. It implicitly learns the conditional generation by inverting the semantic vectors. Such implicit encoding may cause the generated samples to be incoherent with the given attributes. Instead, we explicitly blend the semantic information into each of the coupling layers of the generative flow, learning the semantically consistent visual features. We also argue that the MMD regularization in IZF is inappropriate for GZSL, since the seen and unseen classes are coalesced. It enforces conditional generation by mixing the conditional vectors with the prior. However, this design weakens the conditional information through the affine coupling layers. In our framework, instead, we proactively feed the conditional information into each of the affine coupling layers, which enhances the semantic information in the generation process.
C. Entropy Maximization

Entropy regularization in supervised learning is usually associated with confidence penalty, which is invoked for out-of-distribution detection [35], domain generalization [36], [37], and adversarial samples [38]. By maximizing the entropy of the probability distribution, the prediction confidence for the input sample is penalized. Zhao et al. [36] developed an efficient maximum-entropy regularizer to generate hard adversarial perturbations from information bottleneck principle. Alexmi et al. [35] investigated that variational information bottleneck can improve the classification calibration and the ability to detect out-of-distribution samples. They demonstrate that a deterministic classifier is overconfident, and a high predictive entropy signifies a network does not know how to classify the inputs. These work inspire us to consider mining boundary samples with entropy maximization to explicitly explore the hard samples in zero-shot learning. By learning more extreme visual variants that are close to negative classes, the generalization ability to unseen classes is significantly improved.

III. METHODOLOGY

This section begins with formulating the GZSL problem and introducing the notations. The proposed GSMFlow is outlined next, including the boundary sample mining technique, visual perturbation strategy, the relative positioning approach, and the proactively conditional generative flow to mitigate the generation shifts problem. The model training and zero-shot recognition are introduced lastly.

A. Preliminaries

Consider two datasets - a seen set $S$ and an unseen set $U$. The seen set $S$ contains $N_s$ training samples $x^s \in \mathcal{X}$ and the corresponding class labels $y^s$, i.e., $S = \{x^{s(i)}, y^{s(i)}\}_{i=1}^{N_s}$. Similarly, $U = \{x^u(i), y^u(i)\}_{i=1}^{N_u}$, where $N_u$ is the number of unseen samples. There are $C_s$ seen and $C_u$ unseen classes, so that $y^s \in \{1, \ldots, C_s\}$ and $y^u \in \{C_s + 1, \ldots, C_s + C_u\}$. Note that the seen and unseen classes are mutually exclusive. There are $C_s + C_u$ class-level semantic vectors $a \in \mathcal{A}$, where the class-level semantic vectors for the seen and unseen classes are $\{a_i^{s}\}_{i=1}^{C_s}$ and $\{a_i^{u}\}_{i=1}^{C_u}$, respectively. $a_i$ represents the semantic vector for the $i$-th class. In the setting of GZSL, we only have access to the seen samples $S$ and semantic vectors $\{a_i^{s}\}_{i=1}^{C_s}$ during training. Hence, for brevity, in the demonstration of the training process, we will omit the superscript $s$ for all the seen samples, i.e., $x = x^s$, $a = a^s$.

B. Mining Boundary Samples With Entropy Maximization

To mitigate the variance collapse issue, we propose to explicitly learn the intra-class relationship by mining boundary samples with entropy maximization. We start by learning a contrastive network $g(\cdot)$ with the training data of seen classes. A visual sample $x$ is fused with each semantic vector $a_i$ of seen classes as the visual-semantic pairs, and the corresponding one-hot label is formulated as:

$$\text{OneHot}(x, a_i) = \begin{cases} 0, & y(x) \neq y(a_i) \\ 1, & y(x) = y(a_i) \end{cases},$$

(1)

where $y(\cdot)$ denotes the class label of the input. When the labels of visual and the semantic pairs are matched, we assign 1 to the index of this pair. The contrastive network takes input from the visual-semantic pairs and produces a matching probability between 0 and 1 with a Sigmoid activation function. The classification loss function for the contrastive network can then be formulated as:

$$\mathcal{L}_{con} = \sum_{i=1}^{C_s} \|g(x, a_i) - \text{OneHot}(x, a_i)\|^2,$$

(2)

where applies a mean squared error between the matching probability of each pair and the one-hot label.

The aim of mining the boundary samples is to perturb the underlying data distribution so that the predictive uncertainty of the contrastive model is enlarged. Given a visual sample $x$ paired with each semantic vector of seen classes, the learned contrastive network $g(\cdot)$ will produce a probability vector over seen classes. This motivates us to involve prediction entropy $H(x)$ of all possible classes:

$$H(x) = \sum_{i=1}^{C_s} g(x, a_i) \log g(x, a_i).$$

(3)

To mine the boundary samples, we fixed the weights in the contrastive network and take the training samples as the learning parameters instead. By maximizing the prediction entropy while still minimizing the classification loss function, the training samples move towards the decision boundaries against negative classes. The update step can be defined as:

$$x^{k+1} = x^k + \eta \nabla_x (\mathcal{L}_{con} - \lambda_1 H(x)),$$

(4)

where $\eta$ represents the $k$-th step in total $K$ steps of the training. The perturbed visual samples in the final step are denoted as $x^+$ and appended to the original dataset.

To further diversify the visual samples, we propose the visual perturbation strategy by adding a small amount of noise to these visual features. We begin with sampling a perturbation vector $e \in \mathbb{R}^{d_v}$ from a Gaussian distribution with the same size $d_v$ as the visual features, $e \sim \mathcal{N}(0, I)$. A diversified sample can be then yielded by:

$$x^+ = x^k + \lambda_2 e,$$

(5)

where $\lambda_2$ is the coefficient that controls the degree of perturbation.

While the visual perturbation reveals more diversity, we may unexpectedly incorporate some noisy samples resulting in the distribution shifts. We argue that the prototype of each class should be invariant when introducing diversified samples. Thus, in order to avoid distribution shift, we aim to fix the prototype for each class when perturbing the real samples. The class prototype for $c$-th class is defined as the mean vector of the samples in the
same class from the empirical distribution $p_X$:
\[
E_{x_c \sim p_X} [\tilde{x}_c] = \frac{1}{N_c} \sum_{i=1}^{N_c} x_c^i,
\]
where $N_c$ is the sample number of the $c$-th class in the training set.

When generating a visual sample conditioned on $c$-th class with the corresponding semantic vector $a_c$, the expected mean sample should be close to the class prototype given the prior $z$ as the mean vector from the distribution, i.e., all zeros $0$:
\[
\mathcal{L}_\text{proto} = \frac{1}{C_c} \sum_{c=1}^{C_c} \| f^{-1}(0, a_c) - E_{x_c \sim p_X} [\tilde{x}_c] \|^2,
\]
where $C_c$ is the number of seen classes.

### C. Relative Positioning

To preserve the geometric information between different attribute categories in the shared subspace, we introduce the relative positioning technique. Specifically, instead of representing the class-level semantic information using the raw attributes, we determine three semantic anchors to correspondingly obtain the relative position of each class in the attribute space. Such a relative position is defined as a global semantic vector, which could better aligns the semantic space with the visual one.

The process of determining the three semantic anchors is described as follows. We begin with constructing a semantic graph with the class-level semantic vectors. The edges $\mathcal{E}$ are defined as the cosine similarities between all semantic vectors:
\[
\mathcal{E}_{ij} = \frac{a_i \cdot a_j}{\| a_i \|_2 \cdot \| a_j \|_2},
\]
where $\mathcal{E}_{ij}$ refers to the similarity between $i$-th class and $j$-th class. Then, for each class, we calculate the sum similarities to all other classes:
\[
d_i = \sum_{j=1}^{C_s} \mathcal{E}_{ij},
\]
where $C_s$ denotes the total number of seen classes. We define the three semantic anchors $a_{\text{max}}$, $a_{\text{min}}$, $a_{\text{med}}$ with the highest, lowest, and median sum similarities to other semantic vectors. The global semantic vectors are then acquired by computing the responses from these three semantic anchors.

The dimensionality of visual features is usually much higher than that of semantic vectors, e.g., 2,048 vs. 85 in the AWA dataset. Thus, the generation process can be potentially dominated by visual features. In Section IV-H, we discuss the impact of the semantic vectors’ dimensionality. To avoid this issue, we apply three functions $h_{\text{max}}(\cdot)$, $h_{\text{min}}(\cdot)$, $h_{\text{med}}(\cdot)$ to map the semantic responses to a higher dimension through the implementation with an FC layer and a ReLU activation function. The global semantic vector for each class can then be formulated as:
\[
a^g_i = h_{\text{max}}(a_i - a_{\text{max}}) + h_{\text{min}}(a_i - a_{\text{min}}) + h_{\text{med}}(a_i - a_{\text{med}}),
\]

The global semantic vectors revised by relative positioning are fed into the conditional generation flow as the conditional information.

### D. Conditional Generative Flow

To model the distribution of unseen visual features, we resort to normalizing flow with conditional information, i.e., conditional generative flow [34], a stream of simple yet powerful generative models.

A conditional generative flow learns inference and generation within the same network. Let $z \in \mathcal{Z}$ be a random variable from a particular prior distribution $p_Z$, e.g., Gaussian, which has the same dimension as the visual features. We denote the inference process as the forward transformation $f(\cdot) : \mathcal{X} \times \mathcal{A} \rightarrow \mathcal{Z}$, whose inverse transformation $f^{-1}(\cdot) : \mathcal{Z} \times \mathcal{A} \rightarrow \mathcal{X}$ is the generation process. The transformation is composed of $L$ bijective transformations $f(\cdot) = f_1(\cdot) \circ f_2(\cdot) \circ \ldots \circ f_L(\cdot)$, the forward and inverse computations are the composition of the $L$ bijective transformations. Then, we can formalize the conditional generative flow as:
\[
z = f(x; a, \theta), \quad z \sim \mathcal{N}(0, I),
\]
where the transformation $f(\cdot)$ parameterized with $\theta$ learns to transform a sample $x$ in the target distribution to the prior data distribution $p_Z$ conditioned on the corresponding class-level semantic vector $a$. The forward transformation has its inverse transformation $f^{-1}(\cdot)$, which flows from the prior distribution $p_Z$ towards the target data distribution $p_X$:
\[
x = f^{-1}(z; a, \theta).
\]

With the change of variable formula, the bijective function $f(\cdot)$ can be trained through maximum log-likelihood:
\[
\log p_X (x; a, \theta) = \log p_Z (f(x; a, \theta)) + \log \left| \det \left( \frac{\partial f}{\partial x} \right) \right|,
\]
where the latter half term denotes the logarithm of the determinant of the Jacobian matrix.

According to Bayes’ theorem, the posterior distribution $p_X(\theta; x, a)$ for the parameter $\theta$ is proportional to $p_X(x; a, \theta) p_\theta(\theta)$. The objective function can then be formulated as:
\[
\mathcal{L}_\text{flow} = \mathbb{E} \left[ - \log p_X (x; a, \theta) - \log p_\theta(\theta) \right],
\]

The composed bijective transformations are $L$ conditional affine coupling layers. As shown in Fig. 4, in the forward transformation, each layer splits the input vector $u$ into two factors $[u_1, u_2]$. Note that in the first coupling layer, the input is $x$. Within each coupling layer, the internal functions $s_1(\cdot)$, $s_2(\cdot)$, and $t_1(\cdot)$, $t_2(\cdot)$ are formulated as:
\[
v_1 = u_1 \odot \exp (s_1 ([u_2, a])), \quad t_1([u_2, a]),
\]
\[
v_2 = u_2 \odot \exp (s_2 ([v_1, a])) + t_2 ([v_1, a]),
\]
where $\odot$ is the element-wise multiplication and the outputs $v_1$ and $v_2$ are fed into the next affine transformation. In the last coupling layer, the output will be $x$. In the inverse direction, the
Conditional affine coupling module takes $v_1$ and $v_2$ as inputs:

$$
\begin{align*}
    u_2 &= (v_2 - t_2 ([v_1, a])) \odot \exp (s_2 ([v_1, a])) , \\
    u_1 &= (v_1 - t_1 ([u_2, a])) \odot \exp (s_1 ([u_2, a])),
\end{align*}
$$

(16)

where $\odot$ is the element-wise division.

As proposed in Real NVP [39], when combining coupling layers, the Jacobian determinant remains tractable and its logarithm is the sum of $s_1(\cdot)$ and $s_2(\cdot)$ over visual feature dimensions.

### E. Training and Zero-Shot Inference

We have demonstrated the strategies for improving visual and semantic spaces, and also the conditional generative flow. In this subsection, we will illustrate the training strategy and zero-shot inference stage.

It is worth mentioning that GSMFlow framework involves two training stages. First of all, in the boundary sample mining stage, a contrastive network $g(\cdot)$ is trained with the loss function defined in (2). Then, we free the contrastive network and propagate the original visual samples according to (4). The original samples after prerogation for a particular number of steps are the mined boundary samples $x^+$. With the derived boundary samples $x^+$ and the global semantic vectors $a^g$, the objective functions $L_{\text{flow}}$ and $L_{\text{proto}}$ in (14) and (7) should be rewritten as:

$$
\begin{align*}
    L_{\text{flow}} &= \mathbb{E} \left[ - \log p_X (x^+; a^g, \theta) \right] - \log (p_0(\theta)), \\
    L_{\text{proto}} &= \frac{1}{C_s} \sum_{c=1}^{C_s} \left\| f^{-1}(0, a^g) - \mathbb{E}_{x \sim -p_X} [x_c] \right\|^2. 
\end{align*}
$$

(17)

Then, the overall objective function of the proposed GSMFlow is formulated as:

$$
\min_{\theta} (L_{\text{flow}} + \lambda_3 L_{\text{proto}}),
$$

(18)

where $\lambda_3$ is the coefficient of the prototype loss.

After the conditional generative flow is trained on the seen classes with the boundary samples and the global semantic vectors, it is leveraged to generate visual features of unseen classes:

$$
x^u = f^{-1}(z; a^g, \theta).
$$

(19)

A softmax classifier is then trained on the real visual features of seen classes and the synthesized visual features of unseen classes. For the coming test samples from either seen or unseen classes, the softmax classifier aims to predict the corresponding class label accurately.

### IV. Experiments

In this section, we evaluate our approach GSMFlow in both generalized zero-shot learning and conventional zero-shot learning tasks. We first introduce the datasets and experimental settings and then compare GSMFlow with the state-of-the-art methods. Finally, we study the effectiveness of the proposed model with a series of ablation study and hyper-parameter sensitivity analysis.

#### A. Datasets

We conduct experiments on four widely used benchmark datasets of image classification. They are two fine-grained datasets, i.e., Caltech-UCSD Birds-200-2011 (CUB) [50] and Oxford Flowers (FLO) [51], two coarse-grained datasets, i.e., Attribute Pascal and Yahoo (aPaY) [52] and Animals with Attributes 2 (AWA) [53]. CUB consists of 11,788 images from 200 fine-grained bird species, in which 150 selected as seen classes and 50 as unseen classes. For FLO, it contains 8,189 images from 102 flower categories, 82 of which are chosen as seen classes. The class-level semantic vectors of CUB and FLO are extracted from the fine-grained visual descriptions (10 sentences per image), yielding 1,024-dimensional character-based CNN-RNN features [45] for each class. aPaY dataset contains 18,627 images from 42 classes. There are 30 seen classes and 12 unseen classes respectively. Each class is annotated with 64 attributes. AWA2 is a considerably larger dataset with 30,475 images from 50 classes and they are annotated with 85 attributes. The statistics and the image exemplars of the four datasets are presented in Table I and Fig. 5, respectively.

| Dataset   | aPaY | AWA | CUB | FLO |
|-----------|------|-----|-----|-----|
| # Attributes | 64   | 85  | 1024| 1024|
| # Seen Classes | 20   | 40  | 150 | 82  |
| # Unseen Classes | 12   | 10  | 50  | 20  |
| # Total Images  | 18,627| 30,475| 11,788| 8,189|

Following the compared methods, we used ResNet101 pre-trained on ImageNet1k without finetuning to extract the 2048d visual features from the entire images. For the split, 40 of the total classes are selected as seen classes and ten as unseen classes. For each dataset, We follow the proposed split setting in [55] and [51].
B. Implementation Details

Our framework is implemented with the open-source machine learning library PyTorch. The conditional generative flow consists of a series of affine coupling layers. Each affine coupling layer is implemented with two fully connected (FC) layers and the first FC layer is followed by a LeakyReLU activation function. The hidden dimension of the FC layer is set as 2,048. The coefficients \( \lambda_1 \) of perturbation degree and the coefficient \( \lambda_2 \) of the prototype loss are set within \{0.02, 0.05, 0.15, 0.3, 0.5\} and \{1, 3, 10, 20, 30\}. The dimension of global semantic vectors varies in \{128, 256, 512, 1,024, 2,048\} and the number of the affine coupling layers varies in \{1, 3, 5, 10, 20\}. The corresponding results are given in Section IV-H. The function \( h(\cdot) \) is implemented with an FC layer and a ReLU activation function. The contrastive network \( g(\cdot) \) for mining the boundary samples is implemented by two linear layers, following with a ReLU activation function and a Sigmoid activation function, respectively. We use Adam optimizer with \( \beta_1 = 0.9 \), \( \beta_2 = 0.999 \) and set the batch size to 256 and set the learning rate to 3e-4. All the experiments are performed on a Lenovo workstation with two NVIDIA GeForce GTX 2080 Ti GPUs.

C. Evaluation Metrics

To avoid the failure of classification accuracy for imbalanced class distributions, we adopt average per-class Top-1 accuracy as the fair evaluation criteria for conventional ZSL and the seen and unseen set performance in GZSL:

\[
\text{Acc}_Y = \frac{1}{|Y|} \sum_{y=1}^{|Y|} \frac{\# \text{ of correct predictions in } y}{\# \text{ of samples in } y},
\]

(20)

where \( |Y| \) is the number of testing classes. A correct prediction is defined as the highest probability of all candidate classes. Following [55], the harmonic mean of the average per-class Top-1 accuracies on seen \( \text{Acc}_{S} \) and unseen \( \text{Acc}_{U} \) classes are used to evaluate the performance of generalized zero-shot learning. It is computed by:

\[
H = \frac{2 \times \text{Acc}_{S} \times \text{Acc}_{U}}{\text{Acc}_{S} + \text{Acc}_{U}}.
\]

(21)

D. Comparisons With State-of-The-Art Methods

Table II summarizes the performance comparison between our proposed GSMFlow and the other state-of-the-art methods in the setting of GZSL. We choose the most representative state-of-the-art methods for comparison, the top six methods marked with \( \dagger \) are embedding-based methods, while the below eleven methods and our proposed method marked with \( \ddagger \) are generative methods. The unseen, seen and the balanced harmonic mean between the two domains are represented as U, S, and H. It can be seen that our proposed framework consistently outperforms other methods except on CUB dataset. For the semantic information of CUB in the compared methods, the recent methods E-PGN, CE-GZSL and SDGZSL also use the 1,024-dimensional semantic information as in this work. It is worth noting that OOD-ZSL achieves the best performance on CUB as it leverages a specific calibration technique to threshold the confidence of the prediction being seen or unseen. The thresholding hyperparameter is sensitive to the datasets, which makes it hard to generalize when more unseen classes are involved. Among the generative methods, IZF [18] also leverages normalizing flows as the base generative model. However, we show that by mitigating the generation shifts, our proposed GSMFlow achieves significant improvements on these four datasets. In addition to the conference version, the proposed boundary sample mining technique with entropy maximization introduces more diversified visual potentials to enrich the visual space. In result, we achieve 1.1%, 0.5%, 1.8% and 1.1% higher on the four datasets respectively.

E. Conventional Zero-Shot Learning

Even if GSMFlow is mainly proposed for GZSL, to further validate the effectiveness of our proposed framework, we also conduct experiments in the context of conventional ZSL. Table II summarizes the conventional zero-shot learning performance, which is reported under the T1 metric. We achieve better performance than all the compared methods on the aPaY, CUB, and FLO datasets. For the AWA dataset, we can also achieve comparatively good performance. It can be seen that even if IZF achieves the best performance on AWA dataset with the...
TABLE II

| Methods         | aPaY | AWA | CUB | FLO |
|-----------------|------|-----|-----|-----|
|                 | T1   | U   | S   | H   | T1   | U   | S   | H   | T1   | U   | S   | H   |
| LATEM [12]      | 35.2 | 0.1 | 73.0| 0.2 | 55.8 | 13.3| 77.3| 20.0| 49.3 | 15.2| 57.3| 24.0|
| ALE [11]        | 39.7 | 4.6 | 73.7| 8.7 | 62.3 | 14.0| 81.8| 23.9| 54.9 | 23.7| 62.8| 34.4|
| SIE [13]        | 31.7 | 1.3 | 71.4| 2.6 | 61.9 | 8.0 | 73.1| 11.6| 54.0 | 23.5| 59.2| 33.6|
| ZAE [36]        | 8.3  | 0.4 | 80.9| 0.9 | 84.1 | 1.1 | 82.2| 2.2 | 33.9 | 7.8 | 54.0| 13.6|
| TPN [52]        | 38.9 | 24.1| 64.0| 35.1| 71.2 | 61.2| 65.8| 63.4| 59.5 | 52.6| 52.0| 52.3|
| DVB [33]        | -    | 32.6| 58.3| 41.8| -    | 63.6| 70.8| 67.0| -    | 53.2| 60.2| 56.5|
| GZSL [54]       | 41.1 | 14.2| 78.6| 24.0| 70.2 | 35.4| 86.9| 50.3| 55.8 | 31.7| 61.3| 41.8|
| f-CLSWGAR [18]  | 40.5 | 32.9| 61.7| 42.9| 65.3 | 56.1| 65.5| 60.4| 57.3 | 43.7| 57.7| 49.7|
| CANZSL [37]     | -    | -   | -   | -   | 68.9 | 49.7| 70.2| 58.2| 60.6 | 47.9| 58.1| 52.5|
| GADA-VAE [17]   | 30.3 | 31.7| 55.1| 40.3| 64.0 | 55.8| 75.0| 63.9| 60.4 | 51.6| 53.5| 52.4|
| f-VAEGAN-D2 [55]| -    | -   | -   | -   | 71.1 | 57.6| 70.6| 63.5| 61.0 | 48.4| 60.1| 53.6|
| Tf-VAEGAN [56]  | -    | -   | -   | -   | 72.2 | 59.8| 75.1| 66.6| 64.9 | 52.8| 64.7| 58.1|
| E PGN [37]      | -    | -   | -   | -   | 73.4 | 52.6| 83.5| 64.6| 72.4 | 52.0| 61.1| 56.2|
| IZF [29]        | 44.9 | 42.3| 60.5| 49.8| 74.5 | 60.6| 77.5| 68.0| 67.1 | 52.7| 68.0| 59.4|
| OOD-GZSL [58]   | -    | -   | -   | -   | 55.9 | 94.9| 70.3| 53.8 | 94.6| 68.6|
| CE-GZSL [59]    | -    | -   | -   | -   | 70.4 | 63.1| 78.6| 70.0| 77.5 | 63.9| 68.8| 65.3|
| FREE [60]       | -    | -   | -   | -   | 60.4 | 75.4| 67.1| -   | 55.7 | 59.9| 57.7| -   |
| SDGZSL [61]     | 45.4 | 38.0| 57.4| 45.7| 72.1 | 64.6| 73.6| 68.8| 75.5 | 59.9| 66.4| 63.0|

For GZSL, results are reported in terms of Top-1 accuracy of unseen (U) and seen (S) classes, together with their harmonic mean (H). The best results of the harmonic mean are highlighted in bold. † and ‡ represent embedding-based and generative methods, respectively. ∗ indicates the performance results of method in our conference version. For this extension, we report the average performance results and the standard deviations from multiple experiments with different random seeds.

For ZSL, Performance Results are Reported With the Average Top-1 Classification Accuracy (T1).

For conventional zero-shot learning setting, i.e., only considering the unseen classes during inference. Our proposed GSMFlow shows a significant advantage on the generalized zero-shot learning setting, where both seen and unseen classes are considered. Specifically, GSMFlow surpass IZF by 5.9%, 2.8% and 4.8% on U, S and H respectively. Comparing with the performance results reported in our conference version, the proposed boundary sample mining technique boosts the conventional ZSL performance on the four datasets by 1.5%, 0.5%, 1.9%, and 3.7% respectively.

**F. Class-Wise Performance Comparison**

To further investigate the performance boost through boundary samples and visual perturbation, in Fig. 6, we compare the class-wise accuracy between

- (a) GSMFlow complete,
- (b) GSMFlow w/o EM,
- (c) GSMFlow w/o EM & VP

by illustrating the confusion matrices in the three settings on aPaY dataset. Comparing between (b) and (c), when perturbing the visual features, the generator is exposed to more diverse visual samples, resulting in better generalization ability to unseen classes. Especially, in this case study, the horses and the sheep are the two unseen classes in aPaY dataset. We can notice that horses can be easily misclassified as sheep and can only achieve 14% accuracy. The performance surges to 55% when we introduce the visual perturbation strategy. Similar observations hold for other classes. We further compare class-wise performance between with and without mining boundary samples. It can be seen that the performance improvement mostly come from three classes, i.e., donkey, goat, and jetski. By learning from boundary samples, we surpass the model in the conference version by 10%, 24% and 7% respectively.
TABLE III  

| Methods                  | U  | S   | H  | U  | S   | H  | U  | S   | H  | U  | S   | H  | U  | S   | H  | U  | S   | H  |
|--------------------------|----|-----|----|----|-----|----|----|-----|----|----|-----|----|----|-----|----|----|-----|----|
| IZF w/o constraints      | 35.2 | 54.2 | 42.7 | 38.1 | 78.9 | 51.4 | 48.6 | 54.3 | 51.3 | 59.3 | 78.2 | 67.5 |
| GSMFlow w/o constraints  | 36.5 | 60.0 | 45.4 | 53.3 | 67.6 | 59.6 | 52.6 | 61.2 | 56.6 | 68.6 | 80.1 | 73.9 |
| GSMFlow w/o EM&VP        | 38.4 | 62.3 | 47.6 | 62.2 | 71.3 | 66.4 | 57.9 | 65.2 | 61.3 | 81.3 | 83.8 | 82.5 |
| GSMFlow w/o EM&RP        | 39.6 | 61.3 | 48.1 | 62.9 | 80.4 | 70.6 | 56.8 | 66.2 | 61.1 | 80.9 | 86.2 | 83.4 |
| GSMFlow w/o EM           | 42.0 | 62.3 | 50.2 | 64.5 | 82.1 | 72.3 | 61.4 | 67.4 | 64.3 | 86.6 | 87.8 | 87.2 |
| GSMFlow w/o VP           | 39.0 | 64.8 | 48.7 | 64.8 | 81.2 | 72.1 | 62.1 | 69.2 | 65.4 | 86.5 | 88.8 | 87.6 |

G. Ablation Study

To analyze the contribution of each proposed component and the merit from addressing the three problems in generative zero-shot learning, i.e., semantic inconsistency, variance decay, and structure disorder, we conduct an ablation study on the proposed GSMFlow. We decompose the complete framework into six variants. These include: IZF w/o constraints - the conditional generative flow adopted in compared method [18]; GSMFlow w/o constraints - the generative flow in our proposed framework without visual perturbation and global semantic learning; GSMFlow w/o EM&VP - the complete framework without boundary samples and visual perturbation; GSMFlow w/o EM&RP - the original class-level semantic vectors are used and the boundary samples are not introduced; GSMFlow w/o EM - the boundary samples are not introduced (the conference version); and GSMFlow w/o VP - the boundary samples are introduced but the visual perturbation is removed.

Comparing between IZF w/o constraints and GSMFlow w/o constraints, we can see the performance comparison between the two ways of incorporating conditional information in the generative flow. It can be seen that, instead of mixing the conditional information with the prior input in IZF, our explicit conditional strategy that progressively injects the semantic information into the affine coupling layers during training can achieve higher accuracy on GZSL. The results on the variants GSMFlow w/o EM&VP shows that by mitigating the structure disorder issue, the relative positioning strategy that captures geometric relationships between semantic vectors can significantly improve the GZSL performance. GSMFlow w/o EM&RP indicates that effectiveness of the visual perturbation strategy to prevent the variance collapse. GSMFlow w/o EM represents the method in the conference version without boundary samples mining. It can be seen that when VP and RP are both applied, significant improvements are observed. GSMFlow w/o VP shows the effectiveness of the visual perturbation strategy. It can be seen from Table III that when removing the visual perturbation component, the performance results are inferior to the complete framework. This variant setting also demonstrates that the boundary samples and diverse samples are complementary to each other on improving the zero-shot learning performance.

H. Hyper-Parameter Analysis

GSMFlow mainly involves six hyper-parameters in the boundary sample mining and the model training, as shown in Fig. 7, including the entropy weight \( \lambda_1 \), perturbation weight \( \lambda_2 \), prototype loss weight \( \lambda_3 \), dimension of the global semantic vectors, number of the conditional affine coupling layers and the entropy maximization. Varying the entropy weight from 0.1 to 20 for deriving the boundary samples, in Fig. 7(a), we find that AWA and CUB both reach the best performance at 20. aPaY and FLO achieve the best performance at 10 and 1, respectively. The entropy weight controls the smoothness of the probability vector. A higher weight for the entropy loss regularizes the probability vector to be more flat. As different flower categories in FLO are visually similar, a small weight for entropy loss is enough to derive the boundary samples. It turns out that the perturbation weight also varies for different datasets, as shown in Fig. 7(b), the accuracies of aPaY, CUB, AWA, and FLO reach the peak at around 0.02, 0.05, 0.15 and 0.15. There is an overall trend on all datasets that the performance decreases with larger perturbation weights. A small perturbation weight could explore more diverse samples. However, when the perturbation weight is too large, the features could become indistinguishable between other classes. In general, a small perturbation weight can help to achieve better zero-shot learning performance results. The weight of the prototype loss \( \lambda_3 \) does not have a significant impact on the performance. In Fig. 7(c), we vary \( \lambda_3 \) between 1 and 30, and observe that at 20, 1, 10, and 3, the best performance results are achieved on aPaY, CUB, AWA and FLO. As discussed in Section III-C, the semantic vectors usually have lower dimensions than the visual features, which makes the visual features dominate the generation process in the affine coupling layers. In Fig. 7(d), we report the impact from the dimensionality of the semantic vectors. It can be seen that low-dimensional semantic vectors tend to jeopardize the performance. All the best performance results are achieved at 1,024 dimensions. We also investigate the impact on the number of conditional affine coupling layers, which directly influences the generative model size. In Fig. 7(e), we can see the best performance results are from 5 layers for aPaY and 3 layers for other datasets. Lastly, we experiment on the number of steps for boundary sample mining. As shown in Fig. 7(f), updating the training samples for 20 steps is the best for aPaY and FLO and 30 steps for CUB and AWA datasets.

I. t-SNE Visualization

The quantitative results reported for GZSL (Section IV-D) and ZSL (Section IV-E) demonstrate that the visual samples of
the unseen classes generated by GSMFlow are of good quality and effective for classification tasks. To further gain an insight into the boundary samples and the quality of the generated samples for validating the motivation illustrated in Fig. 2, we provide two series of visualization results.

In Fig. 8, we visualize the comparison between the distributions of groundtruth samples without and with boundary samples in (a)(c) and (b)(d) for aPaY and AWA datasets. It can be seen that the derived boundary samples greatly impact the class decision boundaries. In Fig. 8(a) and (c), the inter-class margin is considerably large. After introducing the boundary samples, in Fig. 8(b) and (d) the visual feature space is enriched and inter-class margin is minimized, which represents more extreme visual potentials are explored. The dotted ellipses highlight that the boundary samples that update the decision margin.

We further compare the empirical distribution of the real unseen class data, the synthesized unseen class data by a conditional generative adversarial network (cGAN), our proposed GSMFlow without boundary samples, and our complete model GSMFlow, as depicted in Fig. 9. It can be seen that the distributions of the synthesized unseen data from cGAN suffer
from the generation shifts problem, which is undesirable for approximating optimal decision boundaries. Specifically, each class is collapsed to fixed modes without capturing enough variances. In contrast, thanks to the visual perturbation strategy, the synthesized samples by GSMFlow are much more diverse. The class-wise relationship is also reflected. For example, comparing to other animal species, horse and giraffe share some attribute values. Also, all the marine species, blue whale, walrus, dolphin, and seal are well separated from other animals. Further, in Fig. 9(d) we can see that the margin of decision boundaries are greatly minimized. At the same time, the inter-class decision boundaries are still discriminative. As a result, our generated samples are semantically consistent and more suitable for zero-shot recognition.

V. CONCLUSION

In this paper, we propose a novel Generation Shifts Mitigating Flow (GSMFlow) framework, which is comprised of multiple conditional affine coupling layers for learning unseen data synthesis. The main motivation of the proposed framework is to address three potential distribution shifts in sample generation, i.e., semantic inconsistency, variance collapse, and the structure disorder. In detail, we explicitly blend the semantic information into the transformations in each of the coupling layers, reinforcing the correlations between the generated samples and the corresponding attributes. A visual perturbation strategy is introduced to diversify the generated data, therefore exposing the generative model to more variant visual samples. To avoid structure disorder in the semantic space, we propose a relative positioning strategy to manipulate the attribute embeddings, guiding which to fully preserve the inter-class geometric structure. An extensive suite of experiments and analysis show that GSMFlow outperforms existing generative approaches for GZSL that suffers from the problems of the generation shifts.

REFERENCES

[1] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, “Label-embedding for image classification,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 7, pp. 1425–1438, Jul. 2016.

[2] Y. Xian et al., “Latent embeddings for zero-shot classification,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 69–77.

[3] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele, “Evaluation of output embeddings for fine-grained image classification,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 2927–2936.

[4] Y. Yang et al., “Zero-shot hashing via transferring supervised knowledge,” in Proc. 24th ACM Int. Conf. Multimedia, 2016, pp. 1286–1295.

[5] J. Li et al., “Learning the invariant side of generative zero-shot learning,” in Proc. IEEE/CVF IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 7402–7411.

[6] J. Li et al., “Alleviating feature confusion for generative zero-shot learning,” in Proc. 27th ACM Int. Conf. Multimedia, 2019, pp. 1587–1595.

[7] E. Schonfeld, S. Ebrahimi, S. Sinha, T. Darrell, and Z. Akata, “Generalized zero-shot learning with multiple graph adaptive generative networks,” IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 7, pp. 2903–2915, Jul. 2022.

[8] Y. Ye, T. Pan, T. Luo, J. Li, and H. T. Shen, “Learning modality-consistent latent representations for generalized zero-shot learning,” IEEE Trans. Multimedia, early access, Jan. 26, 2022, doi: 10.1109/TMM.2022.3154257.

[9] Y. Long et al., “From zero-shot learning to conventional supervised classification: Unseen visual data synthesis,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 1627–1636.

[10] Z. Chen, S. Wang, J. Li, and Z. Huang, “Rethinking generative zero-shot learning: An ensemble learning perspective for recognizing visual patches,” in Proc. 28th ACM Int. Conf. Multimedia, 2020, pp. 3413–3421.

[11] Y. Shen et al., “Invertible zero-shot recognition flows,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 4104–4122.

[12] J. Li, M. Jing, K. Lu, L. Zhu, and H. T. Shen, “Investigating the bilateral connections in generative zero-shot learning,” IEEE Trans. Cybern., vol. 52, no. 8, pp. 8167–8178, Aug. 2022.

[13] Z. Chen et al., “Mitigating generation shifts for generalized zero-shot learning,” in Proc. 29th ACM Int. Conf. Multimedia, 2021, pp. 844–852.

[14] I. Tschantzaridis, T. Joachims, T. Hofmann, and Y. Altun, “Large margin methods for structured and interdependent output variables,” J. Mach. Learn. Res., vol. 6, no. 9, pp. 14531484, 2005.

[15] A. Frome et al., “Devise: A deep visual-semantic embedding model,” in Proc. Conf. Neural Inf. Process. Syst., 2013, pp. 2121–2129.

[16] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” in Proc. Conf. Neural Inf. Process. Syst., 2012, pp. 1097–1105.

[17] B. Romera-Paredes and P. Torr, “An embarrassingly simple approach to zero-shot learning,” in Proc. 32nd Int. Conf. Mach. Learn., 2015, pp. 2152–2161.

[18] E. Kodirov, T. Xiang, and S. Gong, “Semantic autoencoder for zero-shot learning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 3174–3183.

[19] Z. Chen, J. Li, Y. Luo, Z. Huang, and Y. Yang, “CANSZSL: Cycle-consistent adversarial networks for zero-shot learning from natural language,” in Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis., 2020, pp. 874–883.

[20] I. Goodfellow et al., “Generative adversarial nets,” in Proc. Conf. Neural Inf. Process. Syst., 2014, pp. 2672–2680.

[21] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in Proc. Int. Conf. Learn. Representations, 2014.

[22] T. Han, Y. Lu, S. Zhu, and Y. Wu, “Alternating back-propagation for generator network,” in Proc. AAAI Conf. Artif. Intell., 2017, vol. 31, no. 1.

[23] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in Proc. Int. Conf. Mach. Learn., 2017, pp. 214–223.

[24] Y. Zhu, J. Xie, B. Liu, and A. Elgammal, “Learning feature-to-feature translator by alternating back-propagation for generative zero-shot learning,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 9844–9854.

[25] J. Zhu, T. Park, P. Isola, and A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 2223–2232.

[26] L. Ardizzone et al., “Analyzing inverse problems with invertible neural networks,” in Proc. Int. Conf. Learn. Representations, 2019.

[27] L. Ardizzone, C. Lüth, J. Kruse, C. Rother, and U. Köthe, “Guided image generation with conditional invertible neural networks,” in Proc. Int. Conf. Learn. Representations, 2019.

[28] A. A. Alemi, I. Fischer, and J. V. Dillon, “Uncertainty in the variational inference bottleneck,” in Proc. Conf. Uncertainty Artif. Intell., UDL Workshop, 2018.

[29] L. Zhao, T. Liu, X. Peng, and D. Metaxas, “Maximum-entropy adversarial data augmentation for improved generalization and robustness,” in Proc. Conf. Neural Inf. Process. Syst., 2020, vol. 33, pp. 14435–14447.

[30] Z. Wang, Y. Luo, R. Qin, Z. Huang, and M. Baktashmotlagh, “Learning to diversify for single-domain generalization,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 834–843.

[31] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, “Deep variational information bottleneck,” in Proc. Int. Conf. Learn. Representations, 2017.
[39] L. Dinh, J. Sohl-Dickstein, and S. Bengio, “Density estimation using real NVP,” in Proc. Int. Conf. Learn. Representations, 2017.

[40] H. Jiang, R. Wang, S. Shan, and X. Chen, “Transferable contrastive network for generalized zero-shot learning,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 9765–9774.

[41] S. Min et al., “Domain-aware visual bias eliminating for generalized zero-shot learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 12664–12673.

[42] Y. Zhu, M. Elhoseiny, B. Liu, X. Peng, and A. Elgammal, “A generative adversarial approach for zero-shot learning from noisy texts,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. 2018, pp. 1004–1013.

[43] Y. Xian, S. Sharma, B. Schiele, and Z. Akata, “F-VAEGAN-D2: A feature generating framework for any-shot learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 10275–10284.

[44] S. Narayan, A. Gupta, F. S. Khan, C. G. Snoek, and L. Shao, “Latent embedding feedback and discriminative features for zero-shot classification,” 2020, arXiv:2003.07833.

[45] Y. Yu, Z. Ji, J. Han, and Z. Zhang, “Episode-based prototype generating network for zero-shot learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 14035–14044.

[46] X. Chen, X. Lan, F. Sun, and N. Zheng, “A boundary based out-of-distribution classifier for generalized zero-shot learning,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 572–588.

[47] Z. Han, Z. Fu, S. Chen, and J. Yang, “Contrastive embedding for generalized zero-shot learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 2371–2381.

[48] S. Chen et al., “FREE: Feature refinement for generalized zero-shot learning,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 122–131.

[49] Z. Chen et al., “Semantics disentangling for generalized zero-shot learning,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 8712–8720.

[50] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, “The caltech-ucsd birds-200-2011 dataset,” California Inst. Technol., Pasadena, CA, USA, Tech. Rep. 2010-001, 2011.

[51] M. E. Nilsback and A. Zisserman, “Automated flower classification over a large number of classes,” in Proc. 6th Indian Conf. Comput. Vis., Graph. & Image Process., 2008, pp. 722–729.

[52] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, “Describing objects by their attributes,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2009, pp. 1778–1785.

[53] C. H. Lampert, H. Nickisch, and S. Harmeling, “Attribute-based classification for zero-shot visual object categorization,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 3, pp. 453–465, Mar. 2014.

[54] S. Reed, Z. Akata, H. Lee, and B. Schiele, “Learning deep representations of fine-grained visual descriptions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 49–58.

[55] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, “Zero-shot learning—A comprehensive evaluation of the good, the bad and the ugly,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 9, pp. 2251–2265, Sep. 2019.

Yadan Luo received the B.Sc. degree in computer science from the University of Electronic Engineering and Technology of China, Sichuan, China, in 2017, and the Ph.D. degree with the University of Queensland (UQ), Brisbane, QLD, Australia, in 2021. She is currently a Postdoctoral Research Fellow with UQ. Her research interests include multimedia retrieval, machine learning, and computer vision.

Sen Wang received the Ph.D. degree from the University of Queensland (UQ), Brisbane, QLD, Australia, in 2014. From 2014 to July 2016, he was an ARC Postdoctoral Research Fellow with UQ. From July 2016 to 2018, he was with the School of Information and Communication Technology, Griffith University, Brisbane, QLD, Australia. In 2019, he joined the School of Information Technology and Electrical Engineering, UQ. He is currently a Senior Lecturer and an ARC DECRA Fellow. His research interests include data mining and machine learning include human action recognition, zero-shot learning, domain adaptation, feature selection, reinforcement learning, time-series analysis, temporal knowledge graphs, and health informatics. He was the recipient of Australia Discovery Early Career Researcher Award.

Jingjing Li received the M.Sc. and Ph.D. degrees in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 2013 and 2017, respectively. He is currently a National Postdoctoral Program for Innovative Talents Research Fellow with the School of Computer Science and Engineering, University of Electronic Science and Technology of China. His research interests include machine learning, including transfer learning, zero-shot learning, and recommender systems.

Zi Huang (Member, IEEE) received the B.Sc. degree from Tsinghua University, China, and the Ph.D. degree in computer science from The University of Queensland, Australia, in 2014. From 2014 to July 2016, he was an ARC Future Fellow with the School of Information Technology and Electrical Engineering, The University of Queensland, Brisbane, QLD, Australia. Her research interests include multimedia search, social media analysis, database, and information retrieval. Her publications have been published in leading conferences and journals, including the ACM Multimedia, ACM Transactions on Multimedia, IEEE Transactions on Knowledge and Data Engineering, ACM TOIS, and ACM Computing Surveys.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.