GSMFlow: Generation Shifts Mitigating Flow for Generalized Zero-Shot Learning

Zhi Chen, Yadan Luo, Sen Wang, Jingjing Li, Zi Huang

Abstract—**Generalized Zero-Shot Learning (GZSL) aims to recognize images from both the seen and unseen classes by transferring semantic knowledge from seen to unseen classes. It is a promising 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 discover and address three potential problems that trigger the generation shifts, *i.e.*, *semantic inconsistency*, *variance collapse*, and *structure disorder*. First, to enhance the reflection of the semantic information in the generated samples, we explicitly embed the semantic information into the transformation in each conditional affine coupling layer. Second, to recover the intrinsic variance of the real unseen features, we introduce a boundary sample mining strategy with entropy maximization to discover more difficult visual variants of semantic prototypes and hereby adjust the decision boundary of the classifiers. Third, a relative positioning strategy is proposed to revise the attribute embeddings, guiding them 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.**

I. INTRODUCTION

TRADITIONAL visual recognition systems have progressed substantially with the help of a massive amount of labeled data and deep learning techniques [1]–[5]. However, it is challenging for these recognition systems to generalize to unseen classes that are unknown during training. Also, the need for the massive amount of labeled data has been a curse when training a deep recognition system. Generally, one may curate a certain amount of data of every class for training a canonical classification model, while the need for data is drastically 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 [6]–[10]. By incorporating side information, *e.g.,* class-level semantic attributes, ZSL transfers semantic-visual relationships from the seen classes to unseen classes without any visual samples. While conventional ZSL aims to recognize

Z. Chen, Y. Luo, S. Wang and Z. Huang are with School of Information Technology & Electrical Engineering, The University of Queensland, Brisbane, QLD 4072, Australia. (e-mails:zhi.chen@uq.net.au, lyadanluol@gmail.com, sen.wang@uq.edu.au, huang@itee.uq.edu.au).

J. Li is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China (email: jjl@uestc.edu.cn).
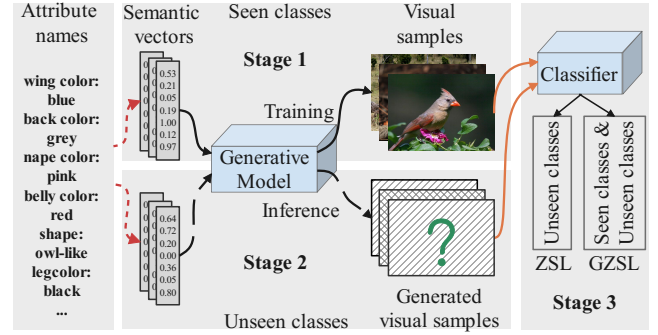

Fig. 1: An illustration of generative zero-shot learning.

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 [11]–[14].

In GZSL, there are two mainstream methods GZSL can be roughly categorized into embedding-based methods [6]–[8], [15], [16] and generative methods [12], [13], [17]–[21]. Embedding-based methods usually cast 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 [12], [13], [22]–[25], as shown in Fig. 1, cast the GZSL problem into a supervised classification task by generating synthesized visual features for unseen classes. Then a supervised classifier can be trained on both real 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 cover 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 samples. Therefore, it is critical to analyze the causes and mitigate the resulted generation shifts. In this paper, as illustrated in Fig. 2, we identify three common types of generation shifts in generative zero-shot learning methods:

- **Semantic inconsistency:** The state-of-the-art GZSL approach [24] focuses on preserving the exact generation cycle consistency by a normalizing flow. It constructs a complex prior and disentangle the outputs into semantic and non-semantic vectors. Such implicit encoding may cause the generated samples to be incoherent with the given attributes and deviated from the real distributions.
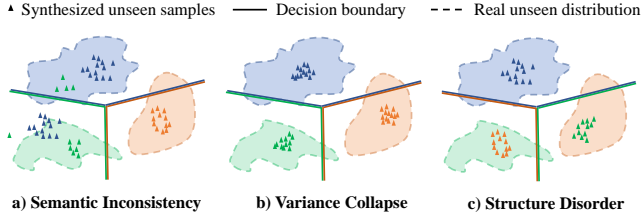
Fig. 2: An illustration of the generation shifts. a) Implicit semantic encoding causes the semantic inconsistency when generating unseen samples. b) The synthesized samples collapse to fixed modes. c) The synthesized samples fail to fully preserve the geometric relationships in the semantic space.

- **Variance collapse:** The synthesized samples commonly collapse into fixed modes and fail to capture the intra-class variance of unseen samples, which can be originated from different poses, illumination and background.
- **Structure disorder:** It is also hard to fully preserve the geometric relationships between different attribute categories in the shared subspace. If the relative position of unseen samples changes in the visual space, the recognition performance will inevitably degenerate.

To address these potential generation shifts, in this paper, we propose a novel framework for generalized zero-shot learning, namely Generation Shifts Mitigating Flow (GSMFlow), as depicted in Fig. 3. In particular, to tackle the semantic inconsistency, we explicitly embed the semantic information into the transformations in each of the coupling layers during inference and generation, enforcing the unseen visual feature generation to be semantically consistent. Furthermore, to mitigate the variance collapse issue, a boundary sample mining technique with entropy maximization is proposed to explicitly discover the decision boundaries between classes and more visual variants that are explored with visual perturbation. With this strategy, the generative model is exposed to more visual variants for each class, which will adjust the decision boundary for suiting 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, then we add a small amount of noise which brings 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 generated 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 explicitly 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 explicit learn the visual representation boundaries for each semantic vector. Moreover, a visual perturbation strategy 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 [26], 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 explicitly 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 2. GSMFlow is presented in Section 3, followed by the experiments and the in-depth analysis in Section 4. Lastly, Section 5 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 [6] proposes to employ embedding functions to measure the compatibility scores between a semantic embedding and a data sample. SJE [8] extends ALE by using structured SVM [27] and takes advantage of the structured outputs. DeViSE [28] constructs a deep visual semantic embedding model to map 4096-dimensional visual features from AlexNet [29] to 500 or 1000-dimensional skip-gram semantic embeddings. EZSL [30] theoretically gives the risk bound of the generalization error and connects zero-shot learning with the domain adaptation problem. More recently, SAE [31] 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 [32]. 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, by generating missing visual samples of unseen classes, zero-shot learning can be converted into a supervised classification task.

B. Generative GZSL

A number of generative methods have been applied for GZSL, *e.g.,* Generative Adversarial Nets (GANs) [33], Variational Autoencoders (VAEs) [34], and Alternating Back-Propagation algorithms (ABPs) [35]. f-CLSWGAN [13] presents a WGAN-based [36] approach to synthesize unseen visual features based on semantic information. CADA-VAE [12] 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 [17] 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 [32] considers the cycle-consistency principle of image generation and proposes a cycle architecture by translating synthesized visual features into semantic information. ABP-ZSL [37] adopts the rarely studied generative models ABPs to generate visual features for unseen classes. GDAN [11] 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 [32], [38], [39]. 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 [40], [41] are proposed to learn image generation from class-level semantic information. IZF [24] adopts the invertible flow model [40] for GZSL. It implicitly learns the conditional generation by inferring 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 progressively input the conditional information in each affine coupling layer, 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 [42], domain generalization [43], [44], and adversarial samples [45]. By maximizing the entropy of the probability distribution, the prediction confidence for the input sample is penalized. Zhao *et al.* [43] developed an efficient maximum-entropy regularizer to generate hard adversarial perturbations from information bottleneck principle. Alexmi *et al.* [42] 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 are significantly improved.

III. METHODOLOGY

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

A. Preliminaries

Consider two datasets - a seen set $\mathcal{S}$ and an unseen set $\mathcal{U}$. The seen set $\mathcal{S}$ contains N^s training samples $\boldsymbol{x}^s \in \mathcal{X}$ and the corresponding class labels y^s, *i.e.,* $\mathcal{S} = \{\boldsymbol{x}^s_{(i)}, y^s_{(i)}\}_{i=1}^{N^s}$. Similarly, $\mathcal{U} = \{\boldsymbol{x}^u_{(i)}, y^u_{(i)}\}_{i=1}^{N^u}$, 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 $\boldsymbol{a} \in \mathcal{A}$, where the class-level semantic vectors for the seen and unseen classes are $\{\boldsymbol{a}^s_i\}_{i=1}^{C_s}$ and $\{\boldsymbol{a}^u_i\}_{i=1}^{C_u}$, respectively. $\boldsymbol{a}_i$ represents the semantic vector for the i-th class. In the setting of GZSL, we only have access to the seen samples $\mathcal{S}$ and semantic vectors $\{\boldsymbol{a}^s_i\}_{i=1}^{C_s}$ 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.,* $\boldsymbol{x} = \boldsymbol{x}^s$, $\boldsymbol{a} = \boldsymbol{a}^s$.

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 $g(\cdot)$ from seen classes. A visual sample $\boldsymbol{x}$ is fused with each semantic vector $\boldsymbol{a}$ of seen classes as the visual-semantic pairs, and the corresponding one-hot label is formulated as:

$$\text{OneHot}(\boldsymbol{x}, \boldsymbol{a}_i) = \begin{cases} 0, & y(\boldsymbol{x}) \neq y(\boldsymbol{a}_i) \\ 1, & y(\boldsymbol{x}) = y(\boldsymbol{a}_i) \end{cases} \quad \boldsymbol{a}_i \in \mathcal{A}^s, \quad (1)$$
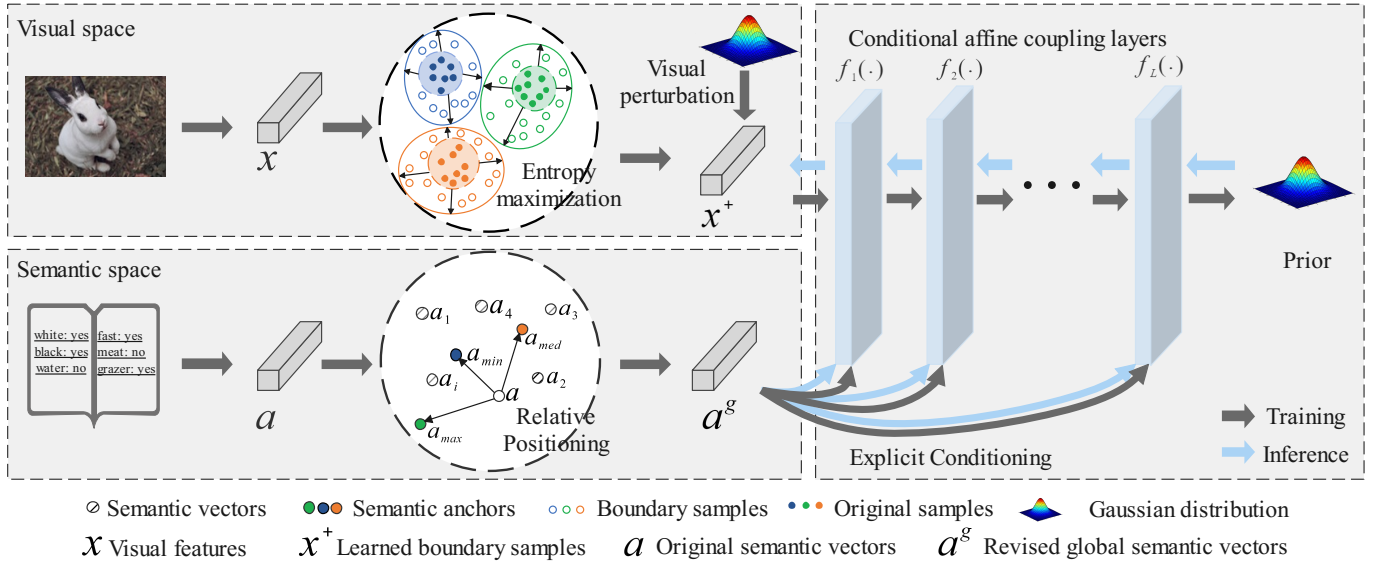
Fig. 3: An illustration of the proposed GSMFlow framework. The conditional generative flow is comprised of a series of conditional affine coupling layers. Particularly, the boundary samples are learned, and additional noise are injected into the visual features to complement the potential patterns. Further, the global semantics are computed with relative positioning to semantic anchors and explicitly encoded into each conditional affine coupling layer. For inference, a latent variable z is inferred from the visual features of an image sample x conditioned on a global semantic vector a^g. Inversely, given z drawn from a prior distribution and a global semantic vector a^g, GSMFlow can generate a visual sample accordingly.

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:

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

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 $\boldsymbol{x}$ 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 $H(\boldsymbol{x})$ of all possible classes:

$$H(\boldsymbol{x}) = \sum_{i=1}^{C_s} g(\boldsymbol{x}, \boldsymbol{a}_i) \log g(\boldsymbol{x}, \boldsymbol{a}_i). \qquad (3)$$

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:

$$\boldsymbol{x}^{k+1} \leftarrow \boldsymbol{x}^k + \eta \bigtriangledown_{\boldsymbol{x}} (\mathcal{L}_{con} - \lambda_1 H(\boldsymbol{x})), \qquad (4)$$

where k represents the k-th step in total K steps of the training. The perturbed visual samples in the final step are denoted as $\boldsymbol{x}^+$ 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 $\boldsymbol{e} \in \mathbb{R}^{d_v}$ from a Gaussian distribution with the same size d_v as the visual features, $\boldsymbol{e} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. A diversified sample can be then yielded by:

$$\boldsymbol{x}^+ = \boldsymbol{x}^+ + \lambda_2 \boldsymbol{e}, \qquad (5)$$

where λ_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 $p_{\bar{X}}$:

$$\mathbb{E}_{\boldsymbol{x}_c \sim p_{\bar{X}}^c}[\bar{\boldsymbol{x}}_c] = \frac{1}{N_c} \sum_{i=1}^{N_c} \boldsymbol{x}_c^i, \qquad (6)$$

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 $\boldsymbol{a}_c$, 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 $\mathbf{0}$:

$$\mathcal{L}_{proto} = \frac{1}{C_s} \sum_{c=1}^{C_s} \|f^{-1}(\mathbf{0}, \boldsymbol{a}_c) - \mathbb{E}_{\boldsymbol{x}_c \sim p_{\bar{X}}^c}[\bar{\boldsymbol{x}}_c]\|^2, \quad (7)$$

where C_s 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:

$$\mathcal{E}_{ij} = \frac{\boldsymbol{a}_i \cdot \boldsymbol{a}_j}{\|\boldsymbol{a}_i\|_2 \cdot \|\boldsymbol{a}_j\|_2}, \quad (8)$$

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:

$$\boldsymbol{d}_i = \sum_{j=1}^{C_s} \mathcal{E}_{ij}, \quad (9)$$

where C_s denotes the total number of seen classes. We define the three semantic anchors $\boldsymbol{a}_{max}$, $\boldsymbol{a}_{min}$, $\boldsymbol{a}_{med}$ 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 $h_{max}(\cdot), h_{min}(\cdot), h_{med}(\cdot)$ 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:

$$\begin{aligned} \boldsymbol{a}_i^g = {}& h_{max}(\boldsymbol{a}_i - \boldsymbol{a}_{max}) + h_{min}(\boldsymbol{a}_i - \boldsymbol{a}_{min}) \\ & + h_{med}(\boldsymbol{a}_i - \boldsymbol{a}_{med}). \end{aligned} \quad (10)$$

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 [41], a stream of simple yet powerful generative models.
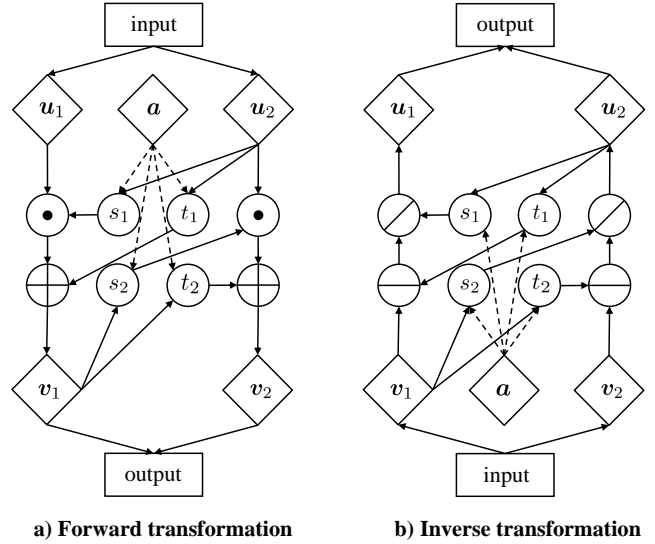

Fig. 4: The flowcharts of the conditional affine coupling layers. (a) The transformation flowchart when training the model. (b) The inverse direction for generating new samples.

A conditional generative flow learns inference and generation within the same network. Let $\boldsymbol{z} \in \mathcal{Z}$ 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 $f(\cdot) : \mathcal{X} \times \mathcal{A} \to \mathcal{Z}$, whose inverse transformation $f^{-1}(\cdot) : \mathcal{Z} \times \mathcal{A} \to \mathcal{X}$ is the generation process. The transformation is composed of L bijective transformations $f(\cdot) = f_1(\cdot) \circ f_2(\cdot) \circ ... \circ f_L(\cdot)$, the forward and inverse computations are the composition of the L bijective transformations. Then, we can formalize the *conditional generative flow* as:

$$\boldsymbol{z} = f(\boldsymbol{x}; \boldsymbol{a}, \theta), \quad \boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \quad (11)$$

where the transformation $f(\cdot)$ parameterized with θ learns to transform a sample $\boldsymbol{x}$ in the target distribution to the prior data distribution p_Z conditioned on the corresponding class-level semantic vector $\boldsymbol{a}$. 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:

$$\boldsymbol{x} = f^{-1}(\boldsymbol{z}; \boldsymbol{a}, \theta). \quad (12)$$

With the *change of variable formula*, the bijective function $f(\cdot)$ can be trained through maximum log-likelihood:

$$\log p_X(\boldsymbol{x}; \boldsymbol{a}, \theta) = \log p_Z(f(\boldsymbol{x}; \boldsymbol{a}, \theta)) + \log \left| \det(\frac{\partial f}{\partial \boldsymbol{x}}) \right|, \quad (13)$$

where the latter half term denotes the logarithm of the determinant of the Jacobian matrix.

According to Bayes' theorem, the posterior distribution $p_X(\theta; \boldsymbol{x}, \boldsymbol{a})$ for the parameter θ is proportional to $p_X(\boldsymbol{x}; \boldsymbol{a}, \theta) p_\theta(\theta)$. The objective function can then be formulated as:

$$\mathcal{L}_{flow} = \mathbb{E}[-\log p_X(\boldsymbol{x}; \boldsymbol{a}, \theta)] - \log p_\theta(\theta), \quad (14)$$

The composed bijective transformations are L conditional affine coupling layers. As shown in Figure 4, in the forward
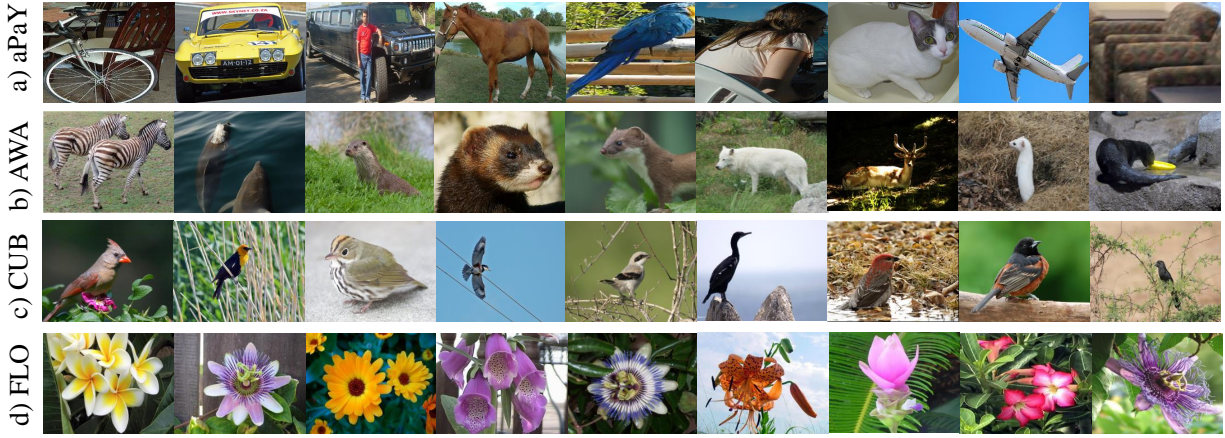
Fig. 5: Exemplars of four datasets, including two coarse-grained datasets (*i.e.,* aPaY and AWA), and two fine-grained datasets (*i.e.,* CUB and FLO).

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:

$$v_1 = u_1 \odot \exp(s_1([u_2, a])) + t_1([u_2, a]),$$
$$v_2 = u_2 \odot \exp(s_2([v_1, a])) + t_2([v_1, a]), \quad (15)$$

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 z. In the inverse direction, the conditional affine coupling module takes v_1 and v_2 as inputs:

$$u_2 = (v_2 - t_2([v_1, a])) \oslash \exp(s_2([v_1, a])),$$
$$u_1 = (v_1 - t_1([u_2, a])) \oslash \exp(s_1([u_2, a])), \quad (16)$$

where $\oslash$ is the element-wise division. As proposed in Real NVP [46], 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 Eq. 2. Then, we free the contrastive network and propagate the original visual samples according to Eq. 4. The original samples after prorogation 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 $\mathcal{L}_{flow}$ and $\mathcal{L}_{proto}$ in Equation 14 and Equation 7 should be rewritten as:

$$\mathcal{L}_{flow} = \mathbb{E}[-\log p_X(x^+; a^g, \theta)] - \log(p_\theta(\theta)),$$
$$\mathcal{L}_{proto} = \frac{1}{C_s} \sum_{c=1}^{C_s} \|f^{-1}(\mathbf{0}, a_c^g) - \mathbb{E}_{x_c \sim p_{\bar{X}}^c}[\bar{x}_c]\|^2. \quad (17)$$

TABLE I: Dataset statistics for aPaY, AWA, CUB and FLO.

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

Then, the overall objective function of the proposed GSM-Flow is formulated as:

$$\min_\theta (\mathcal{L}_{flow} + \lambda_3 \mathcal{L}_{proto}), \quad (18)$$

where λ_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:

$$\bar{x}^u = f^{-1}(z; a^g, \theta). \quad (19)$$

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**) [57] and Oxford Flowers (**FLO**) [58], two coarse-grained datasets, *i.e.,*

TABLE II: Performance comparison in accuracy (%) of the state-of-the-art ZSL and GZSL on four datasets. For ZSL, performance results are reported with the average top-1 classification accuracy (T1). 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.

	Methods	aPaY				AWA				CUB				FLO			
		T1	U	S	H	T1	U	S	H	T1	U	S	H	T1	U	S	H
†	LATEM [7]	35.2	0.1	73.0	0.2	55.8	13.3	77.3	20.0	49.3	15.2	57.3	24.0	60.8	6.6	47.6	11.5
	ALE [6]	39.7	4.6	73.7	8.7	62.5	14.0	81.8	23.9	54.9	23.7	62.8	34.4	48.5	13.3	61.6	21.9
	SJE [8]	31.7	1.3	71.4	2.6	61.9	8.0	73.9	14.4	54.0	23.5	59.2	33.6	53.4	13.9	47.6	21.5
	SAE [31]	8.3	0.4	80.9	0.9	54.1	1.1	82.2	2.2	33.3	7.8	54.0	13.6	-	-	-	-
	TCN [47]	38.9	24.1	64.0	35.1	71.2	61.2	65.8	63.4	59.5	52.6	52.0	52.3	-	-	-	-
	DVBE [48]	-	32.6	58.3	41.8	-	63.6	70.8	67.0	-	53.2	60.2	56.5	-	-	-	-
‡	GAZSL [49]	41.1	14.2	78.6	24.0	70.2	35.4	86.9	50.3	55.8	31.7	61.3	41.8	60.5	28.1	77.4	41.2
	f-CLSWGAN [13]	40.5	32.9	61.7	42.9	65.3	56.1	65.5	60.4	57.3	43.7	57.7	49.7	69.6	59.0	73.8	65.6
	CANZSL [32]	-	-	-	-	68.9	49.7	70.2	58.2	60.6	47.9	58.1	52.5	69.7	58.2	77.6	66.5
	CADA-VAE [12]	42.3	31.7	55.1	40.3	64.0	55.8	75.0	63.9	60.4	51.6	53.5	52.4	65.2	51.6	75.6	61.3
	f-VAEGAN-D2 [50]	-	-	-	-	71.1	57.6	70.6	63.5	61.0	48.4	60.1	53.6	67.7	56.8	74.9	64.6
	TF-VAEGAN [51]	-	-	-	-	72.2	59.8	75.1	66.6	64.9	52.8	64.7	58.1	70.8	62.5	84.1	71.7
	E-PGN [52]	-	-	-	-	73.4	52.6	83.5	64.6	72.4	52.0	61.1	56.2	85.7	71.5	82.2	76.5
	IZF [24]	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 [53]	-	-	-	-	-	55.9	94.9	70.3	-	53.8	94.6	68.6	-	61.9	91.7	73.9
	CE-GZSL [54]	-	-	-	-	70.4	63.1	78.6	70.0	77.5	63.9	66.8	65.3	70.6	69.0	78.7	73.5
	FREE [55]	-	-	-	-	-	60.4	75.4	67.1	-	55.7	59.9	57.7	-	67.4	84.5	75.0
	SDGZSL [56]	45.4	38.0	57.4	45.7	72.1	64.6	73.6	68.8	75.5	59.9	66.4	63.0	86.9	83.3	90.2	86.6
	GSMFlow∗	49.2	42.0	62.3	50.2	72.7	64.5	82.1	72.3	76.4	61.4	67.4	64.3	86.9	86.6	87.8	87.2
	GSMFlow	**50.7** ±1.2	40.9 ±0.6	68.9 ±1.1	**51.3** ±0.7	73.2 ±0.3	66.5 ±0.7	80.3 ±0.7	**72.8** ±0.5	**78.3** ±0.2	63.2 ±0.7	69.3 ±0.7	66.1 ±0.2	**90.6** ±0.5	86.9 ±0.3	89.8 ±0.4	**88.3** ±0.2

Attribute Pascal and Yahoo (**aPaY**) [59] and Animals with Attributes 2 (**AWA**) [60]. 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 [61] 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.

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 [62] and [58].

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 λ_1 of perturbation degree and the coefficient λ_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:

$$Acc_{\mathcal{Y}} = \frac{1}{|\mathcal{Y}|} \sum_{y=1}^{|\mathcal{Y}|} \frac{\# \ of \ correct \ predictions \ in \ y}{\# \ of \ samples \ in \ y}, \quad (20)$$

where $|\mathcal{Y}|$ is the number of testing classes. A correct prediction is defined as the highest probability of all candidate classes. Following [62], the harmonic mean of the average per-class Top-1 accuracies on seen Acc_S and unseen Acc_U classes are used to evaluate the performance of generalized zero-shot learning. It is computed by:

$$H = \frac{2 * Acc_S * Acc_U}{Acc_S + Acc_U}. \quad (21)$$

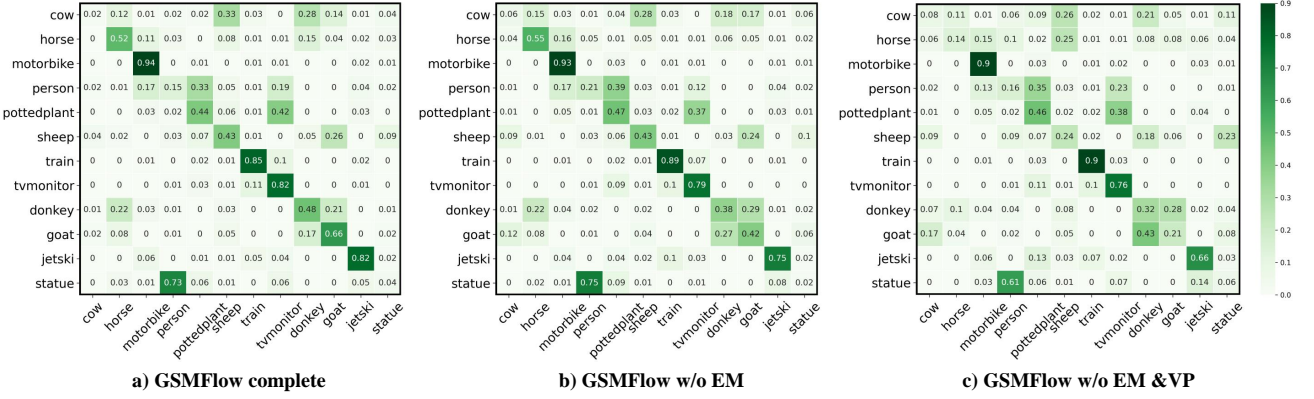**a) GSMFlow complete**   **b) GSMFlow w/o EM**   **c) GSMFlow w/o EM &VP**

Fig. 6: Class-wise performance comparison between with and without visual perturbations. The vertical labels are the groundtruth and the horizontal labels are the predictions.

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 † are embedding-based methods, while the below eleven methods and our proposed method marked with ‡ 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 [24] 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 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 Figure 6, we compare the class-wise accuracy between **a) GSMFlow complete**, **b) GSMFlow w/o EM** and **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.

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 dacay*, 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 [24]; ***GSMFlow w/o constraints*** - the generative

TABLE III: Effects of different components on four datasets. *U*, *S* and *H* represent unseen, seen and harmonic mean, respectively. The best results of harmonic mean are formatted in bold.

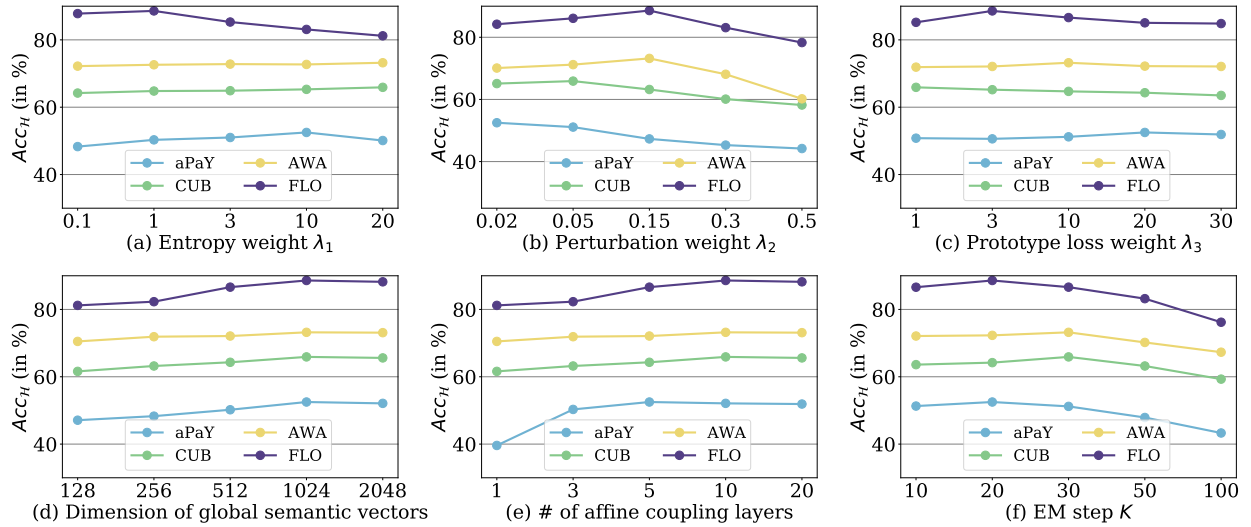| Methods	aPaY			AWA			CUB			FLO		
	U	S	H	U	S	H	U	S	H	U	S	H
IZF w/o constraints	35.2	54.2	42.7	38.1	78.9	51.4	48.6	54.3	51.3	59.3	78.2	67.5
GSMFlow w/o constraints	36.5	60.0	45.4	53.3	67.6	59.6	52.6	61.2	56.6	68.6	80.1	73.9
GSMFlow w/o EM&VP	38.4	62.3	47.6	62.2	71.3	66.4	57.9	65.2	61.3	81.3	83.8	82.5
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
GSMFlow	40.9	68.9	**51.3**	66.5	80.3	**72.8**	63.2	69.3	**66.1**	86.9	89.8	**88.3**

Fig. 7: Hyper-parameter sensitivity. The horizontal axis indicates the varying hyper-parameters for (a) entropy weight, (b) perturbation weight, (c) prototype loss weight, (d) semantic vector dimensions, (e) number of affine coupling layers and (f) the number of entropy maximization steps. The vertical axis reports the corresponding performance.

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. Lastly, when combining these components together, we achieve the best performance results.

H. Hyper-parameter Analysis

GSMFlow mainly involves six hyper-parameters in the boundary sample mining and the model training, as shown in Figure 7, including the entropy weight λ_1, perturbation weight λ_2, prototype loss weight λ_3, dimension of the global semantic vectors, number of the conditional affine coupling layers and

a) Groundtruth features of aPaY

b) Groundtruth & boundary samples of aPaY

c) Groundtruth features of AWA

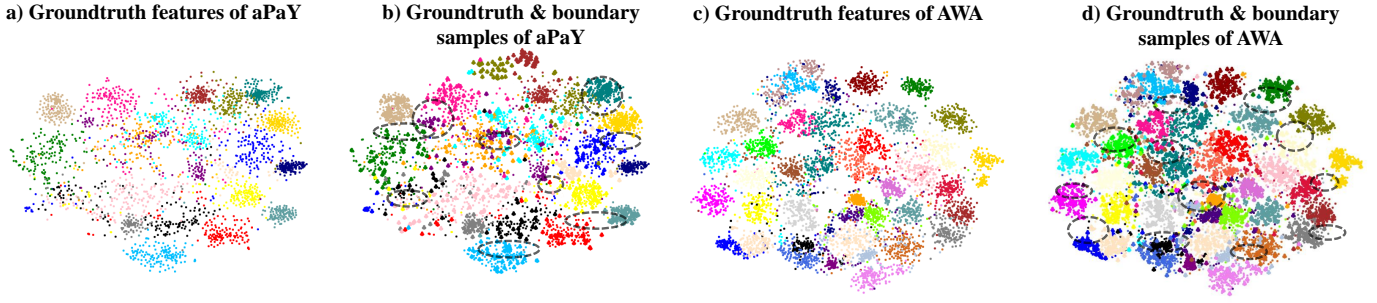**d) Groundtruth & boundary samples of AWA**

Fig. 8: t-SNE visualization of groundtruth features and boundary samples visualization on aPaY and AWA datasets. The dot points represent the groundtruth samples and clubsuit points are the derived boundary samples.

a) Groundtruth features

b) Synthesized features by cGAN
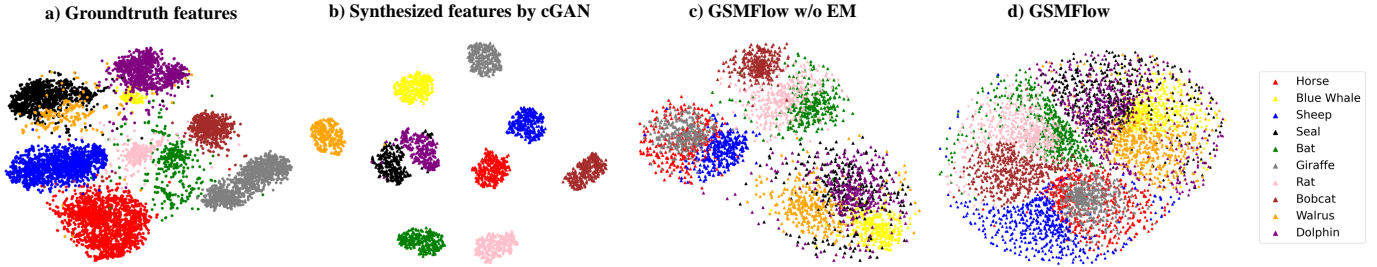
c) GSMFlow w/o EM

d) GSMFlow

Fig. 9: Distribution comparison of the unseen classes in the AWA dataset. (a) The real distributions of the visual features extracted from the backbone. (b) The synthesized distributions by cGAN. (c) The synthesized distributions by GSMFlow w/o EM. (d) The synthesized distribution by the complete GSMFlow model.

the entropy maximization Varying the entropy weight from 0.1 to 20 for deriving the boundary samples, in Figure 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 Figure 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 λ_3 does not have a significant impact on the performance. In Figure 7(c), we vary λ_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 Figure 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 Figure 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 Figure 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 Figure 2, we provide two series of visualization results.

In Figure 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 Figure 8 (a) and (c), the inter-class margin is considerably large. After introducing the boundary samples, in Figure 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 con-

ditional generative adversarial network (cGAN), our proposed GSMFlow without boundary samppls, and our complete model GSMFlow, as depicted in Figure 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 Figure 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] P.-F. Zhang, J. Duan, Z. Huang, and H. Yin, "Joint-teaching: Learning to refine knowledge for resource-constrained unsupervised cross-modal retrieval," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 1517–1525.

[2] P. Zhang, Y. Luo, Z. Huang, X. Xu, and J. Song, "High-order nonlocal hashing for unsupervised cross-modal retrieval," *World Wide Web*, 2021.

[3] P. Zhang, Y. Li, Z. Huang, and X. Xu, "Aggregation-based graph convolutional hashing for unsupervised cross-modal retrieval," *IEEE TMM*, 2021.

[4] J. Dong, Y. Cong, G. Sun, Z. Fang, and Z. Ding, "Where and how to transfer: Knowledge aggregation-induced transferability perception for unsupervised domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[5] J. Dong, Y. Cong, G. Sun, B. Zhong, and X. Xu, "What can be transferred: Unsupervised domain adaptation for endoscopic lesions segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020, pp. 4022–4031.

[6] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-embedding for image classification," *TPAMI*, vol. 38, no. 7, pp. 1425–1438, 2015.

[7] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele, "Latent embeddings for zero-shot classification," in *CVPR*, 2016, pp. 69–77.

[8] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele, "Evaluation of output embeddings for fine-grained image classification," in *CVPR*, 2015, pp. 2927–2936.

[9] Y. Yang, Y. Luo, W. Chen, F. Shen, J. Shao, and H. T. Shen, "Zero-shot hashing via transferring supervised knowledge," in *ACM MM*, 2016.

[10] J. Li, M. Jing, K. Lu, L. Zhu, Y. Yang, and Z. Huang, "From zero-shot learning to cold-start recommendation," in *AAAI*, 2019, pp. 4189–4196.

[11] H. Huang, C. Wang, P. S. Yu, and C. Wang, "Generative dual adversarial network for generalized zero-shot learning," in *CVPR*, 2019, pp. 801–810.

[12] E. Schonfeld, S. Ebrahimi, S. Sinha, T. Darrell, and Z. Akata, "Generalized zero-and few-shot learning via aligned variational autoencoders," in *CVPR*, 2019, pp. 8247–8255.

[13] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, "Feature generating networks for zero-shot learning," in *CVPR*, 2018, pp. 5542–5551.

[14] J. Li, M. Jing, L. Zhu, Z. Ding, K. Lu, and Y. Yang, "Learning modality-invariant latent representations for generalized zero-shot learning," in *ACM MM*, 2020, pp. 1348–1356.

[15] G. Xie, L. Liu, F. Zhu, F. Zhao, Z. Zhang, Y. Yao, J. Qin, and L. Shao, "Region graph embedding network for zero-shot learning," in *ECCV*. Springer, 2020, pp. 562–580.

[16] Y. Li, Z. Liu, L. Yao, and X. Chang, "Attribute-modulated generative meta learning for zero-shot learning," *IEEE TMM*, 2021.

[17] J. Li, M. Jin, K. Lu, Z. Ding, L. Zhu, and Z. Huang, "Leveraging the invariant side of generative zero-shot learning," in *CVPR*, 2019, pp. 7402–7411.

[18] J. Li, M. Jing, K. Lu, L. Zhu, Y. Yang, and Z. Huang, "Alleviating feature confusion for generative zero-shot learning," in *ACM MM*, 2019, pp. 1587–1595.

[19] G. Xie, Z. Zhang, G. Liu, F. Zhu, L. Liu, L. Shao, and X. Li, "Generalized zero-shot learning with multiple graph adaptive generative networks," *IEEE TNNLS*, 2021.

[20] Y. Ye, T. Pan, T. Luo, J. Li, and H. T. Shen, "Learning modality-consistent latent representations for generalized zero-shot learning," *IEEE TMM*, 2022.

[21] H. Su, J. Li, Z. Chen, L. Zhu, and K. Lu, "Distinguishing unseen from seen for generalized zero-shot learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[22] Y. Long, L. Liu, L. Shao, F. Shen, G. Ding, and J. Han, "From zero-shot learning to conventional supervised classification: Unseen visual data synthesis," in *CVPR*, 2017, pp. 1627–1636.

[23] Z. Chen, S. Wang, J. Li, and Z. Huang, "Rethinking generative zero-shot learning: An ensemble learning perspective for recognising visual patches," in *ACM MM*, 2020, pp. 3413–3421.

[24] Y. Shen, J. Qin, L. Huang, L. Liu, F. Zhu, and L. Shao, "Invertible zero-shot recognition flows," in *ECCV*. Springer, 2020, pp. 614–631.

[25] J. Li, M. Jing, K. Lu, L. Zhu, and H. T. Shen, "Investigating the bilateral connections in generative zero-shot learning," *IEEE TCYB*, 2021.

[26] Z. Chen, Y. Luo, S. Wang, R. Qiu, J. Li, and Z. Huang, "Mitigating generation shifts for generalized zero-shot learning," in *ACM Multimedia*, 2021, pp. 844–852.

[27] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *JMLR*, vol. 6, no. 9, 2005.

[28] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov, "Devise: A deep visual-semantic embedding model," in *NeurIPS*, 2013, pp. 2121–2129.

[29] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *NeurIPS*, 2012, pp. 1097–1105.

[30] B. Romera-Paredes and P. Torr, "An embarrassingly simple approach to zero-shot learning," in *ICML*, 2015, pp. 2152–2161.

[31] E. Kodirov, T. Xiang, and S. Gong, "Semantic autoencoder for zero-shot learning," in *CVPR*, 2017, pp. 3174–3183.

[32] Z. Chen, J. Li, Y. Luo, Z. Huang, and Y. Yang, "Canzsl: Cycle-consistent adversarial networks for zero-shot learning from natural language," in *WACV*, 2020, pp. 874–883.

[33] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NeurIPS*, 2014.

[34] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[35] T. Han, Y. Lu, S. Zhu, and Y. Wu, "Alternating back-propagation for generator network," in *AAAI*, vol. 31, no. 1, 2017.

[36] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *ICML*, 2017, pp. 214–223.

[37] Y. Zhu, J. Xie, B. Liu, and A. Elgammal, "Learning feature-to-feature translator by alternating back-propagation for generative zero-shot learning," in *CVPR*, 2019, pp. 9844–9854.

[38] J. Zhu, T. Park, P. Isola, and A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *ICCV*, 2017, pp. 2223–2232.

[39] Z. Chen and Y. Luo, "Cycle-consistent diverse image synthesis from natural language," in *ICMEW*. IEEE, 2019, pp. 459–464.

[40] L. Ardizzone, J. Kruse, S. Wirkert, D. Rahner, E. W. Pellegrini, R. S. Klessen, L. Maier-Hein, C. Rother, and U. Köthe, "Analyzing inverse problems with invertible neural networks," *arXiv preprint arXiv:1808.04730*, 2018.

[41] L. Ardizzone, C. Lüth, J. Kruse, C. Rother, and U. Köthe, "Guided image generation with conditional invertible neural networks," *arXiv preprint arXiv:1907.02392*, 2019.

[42] A. A. Alemi, I. Fischer, and J. V. Dillon, "Uncertainty in the variational information bottleneck," *arXiv preprint arXiv:1807.00906*, 2018.

[43] L. Zhao, T. Liu, X. Peng, and D. Metaxas, "Maximum-entropy adversarial data augmentation for improved generalization and robustness," *NeurIPS*, vol. 33, pp. 14 435–14 447, 2020.

[44] Z. Wang, Y. Luo, R. Qiu, Z. Huang, and M. Baktashmotlagh, "Learning to diversify for single domain generalization," in *ICCV*, 2021, pp. 834–843.

[45] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," *arXiv preprint arXiv:1612.00410*, 2016.

[46] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," *arXiv preprint arXiv:1605.08803*, 2016.

[47] H. Jiang, R. Wang, S. Shan, and X. Chen, "Transferable contrastive network for generalized zero-shot learning," in *ICCV*, 2019, pp. 9765–9774.

[48] S. Min, H. Yao, H. Xie, C. Wang, Z. J. Zha, and Y. Zhang, "Domain-aware visual bias eliminating for generalized zero-shot learning," in *CVPR*, 2020, pp. 12 664–12 673.

[49] Y. Zhu, M. Elhoseiny, B. Liu, X. Peng, and A. Elgammal, "A generative adversarial approach for zero-shot learning from noisy texts," in *CVPR*, 2018, pp. 1004–1013.

[50] Y. Xian, S. Sharma, B. Schiele, and Z. Akata, "f-vaegan-d2: A feature generating framework for any-shot learning," in *CVPR*, 2019, pp. 10 275–10 284.

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

[52] Y. Yu, Z. Ji, J. Han, and Z. Zhang, "Episode-based prototype generating network for zero-shot learning," in *CVPR*, 2020, pp. 14 035–14 044.

[53] X. Chen, X. Lan, F. Sun, and N. Zheng, "A boundary based out-of-distribution classifier for generalized zero-shot learning," in *ECCV*, 2020, pp. 572–588.

[54] Z. Han, Z. Fu, S. Chen, and J. Yang, "Contrastive embedding for generalized zero-shot learning," in *CVPR*, 2021, pp. 2371–2381.

[55] S. Chen, W. Wang, B. Xia, Q. Peng, X. You, F. Zheng, and L. Shao, "Free: Feature refinement for generalized zero-shot learning," in *ICCV*, 2021, pp. 122–131.

[56] Z. Chen, Y. Luo, R. Qiu, S. Wang, Z. Huang, J. Li, and Z. Zhang, "Semantics disentangling for generalized zero-shot learning," in *ICCV*, 2021.

[57] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.

[58] M. E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Indian Conference on Computer Vision, Graphics & Image Processing*, 2008, pp. 722–729.

[59] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *CVPR*, 2009, pp. 1778–1785.

[60] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *TPAMI*, vol. 36, no. 3, pp. 453–465, 2013.

[61] S. Reed, Z. Akata, H. Lee, and B. Schiele, "Learning deep representations of fine-grained visual descriptions," in *CVPR*, 2016, pp. 49–58.

[62] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly," *TPAMI*, pp. 2251–2265, 2018.