Conditional Normalizing Flow-based Generative Model for Zero-Shot Recognition

Haiping Zhang, Xinwei Zhu, Dongjin Yu
School of Computer Science
Hangzhou Dianzi University, HDU
Hangzhou, China
zhanghp, zhuxinwei, yudj@hdu.edu.cn

Liming Guan, Zhongjin Li
School of Information Engineering
Hangzhou Dianzi University, HDU
Hangzhou, China
glm@hdu.edu.cn, lzjhdu@163.com

***Abstract*—Most existing studies on zero-shot recognition（ZSR） are typically about learning a shared embedding space to allow comparison of class prototypes by using nearest-neighbor methods, which suffer from hubness and bias problem. Recent studies attempted to directly synthesize samples of unseen classes by using generative model and have not encountered the aforementioned problems. However, their performance is limited by the inherent problems of VAE and GAN, such as reconstruction loss, mode collapse and unstable training procedure. In this paper, we explore and exploit a novel architecture of the generative model for ZSR, referred to as conditional normalizing flow-based generative model (CNFG). The proposed model consists of a cascade of affine couple transformations and can capture the low-distribution modes of real data density by virtue of its stable and exact log-likelihood maximum training procedure. Extensive experiments and result comparisons of 5 benchmarks have indicated that the normalizing flow-based model is superior to other generative models for ZSR in generalized settings.**

Keywords—zero shot recgnition, generative model, affine couple transformation, hubness problem, model collapse

I. INTRODUCTION

Zero-shot recognition (ZSR) is a learning paradigm that attempts to recognize one object without any (or with zero) annotated data of the object in the training set. Motivated by the learning paradigm of human cognition, ZSR uses auxiliary semantic information of the category to train an effective model, which is required to correctly recognize not only the categories that appear in the training set (seen class) but also those that do not appear in the training set (unseen class). The key point of ZSR is to effectively explore and leverage the semantic knowledge of category that are shared between the seen and unseen class. Early studies on ZSR have mainly focused on the identification of a discriminative semantic representation of categories, such as semantic attributes or word embedding of labels. Leveraging these semantic knowledges, a mapping function from a visual/semantic embedding space to a shared embedding space is learned in the training set and then applied to the testing set. The data distribution of the two domains considerably vary. Thus, several intrinsic problems of the existing mapping-based methods are encountered.

Hubness Problem: [1] has theoretically and empirically demonstrated that hubness curse is an intrinsic characteristic of data distribution density in a high-dimensional space. That is, some hub vector points, which are not similar to other vector points, may be near many other points in a high dimensional space if measured using the nearest-neighbor search methods. The category label in the paradigm of mapping-based ZSR methods is determined by using the nearest-neighbor classifier to identify the most similar class prototype in the shared embedding space. Thus, [2] argue that the hubness problem also severely pollute the existing zero-shot method.

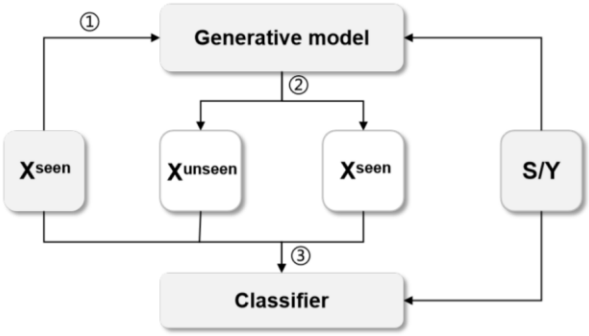

Figure 1. Overall pipeline of the proposed model

Domain Shift and Bias Problem: [3] argue that the mapping function, which is learned from the seen class, is often biased when applied directly to the unseen class because of disjoint classes and the inconsistent manifestation of visual attributes between training data and testing data. This occurrence is referred to as the projection domain shift problem. Coincidentally, [4] also empirically show that the learned mapping function does not perform well in the generalized setting because mapping functions are biased either toward the seen class or the unseen class.

Generative models have recently shown great potential for ZSR and have not encountered the aforementioned problems. These generative model-based ZSR methods attempted to synthesize samples of unseen classes conditioned by the semantic information by using variational autoencoder (VAE) or generative adversarial network (GAN) and cast zero-shot problem as a traditional supervised recognition problem. In this present study, we introduce a novel architecture of the generative model for ZSR, referred to as conditional normalizing flow-

DOI reference number: 10.18293/SEKE2020-093

based generative model (CNFG), which consists of a cascade of affine couple transformations and can capture the low-distribution modes of real data density by virtue of its stable and exact loglikelihood maximum training procedure.

To summarize, the main contribution of this study is threefold.

1) To the best of our knowledge, we are the first to explore and exploit the normalizing flow-based generative model to synthesize unseen data for ZSR problem.

2) We theoretically analyze and derive the formula of our proposed model and argue that our model can capture the low-distribution mode from real data density.

3) We also conduct extensive experiments and comparisons on 5 benchmarks, which shown that the normalizing flow-based generative model is superior to other generative models for ZSR in generalized settings.

II. RELATED WORK

Most existing studies on ZSR are typically about learning a shared embedding space to allow comparison of class prototypes by using nearest-neighbor methods. The pioneering work [5] is the first to propose embedding each class label into the space of attribute vector and cast earlier attribute-based multi-task learning as a label-embedding problem. ESZSL [6] argue that existing approaches to ZSR are highly sophisticated and propose a simple but effective compatibility function to model the relationship between visual embedding and attribute vector by explicitly regularizing the objective function. Prompted by the encoder-decoder paradigm, SAE [7] present a novel architecture consisting of two components. The encoder is responsible for projecting a visual representation into the semantic space similar to most existing ZSR models, and an additional decoder performs the reconstruction from a semantic representation to the visual space. Instead of embedding into a semantic space, DEM [8] propose to use the visual embedding space as the shared embedding space, which less frequently encounters the hubness problem. [9]innovatively introduce graph convolutional network (GCN) for predicting the class visual prototype by using both semantic embedding and the categorical relationships, with semantic embedding as input and visual embedding space as the shared embedding space.

Moreover, generative models have recently shown great potential for ZSR. GAMM [10] compare four different architecture of conditional data generators and emphasize the importance and efficiency of aligning the distributions of real and fake data by using explicit measure metric of distribution divergence, such as the KL divergence and maximum mean discrepancy. CVAE [11] uses the vanilla conditional variational auto-encoder (cVAE) model to directly generate samples conditioned by the given attribute representation for the class. SE-GZSL [12] further introduce a feedback-driven mechanism for cVAE architecture, which is coupled with a multivariate regressor to learn a projection from the cVAE decoder output to the representation of attributes that help increase the discriminative nature of the generated data. f-CLSWGAN [13] enhances Wasserstein GAN by adding a classification loss to the original generator loss for ZSR and enforces the model to generate sufficient image features that are more suitable for training a final multi-modal classifier. Inspired by cycle consistency loss, [14] introduces a multi-modal cycle consistency loss term that enforces better reconstruction from the generated visual representations back to semantic embedding. Using cyclic consistency loss and dual adversarial loss, [15] also proposed a novel model, referred to as GDAN, which combines visual-to-semantic projection, semantic-to-visual projection and a metric learning module in a unified framework and boosts the performance of ZSR. GMN [16] equip a conditional GAN with the gradient matching loss, which can measure the quality of the gradient signal acquired from the synthesized samples.

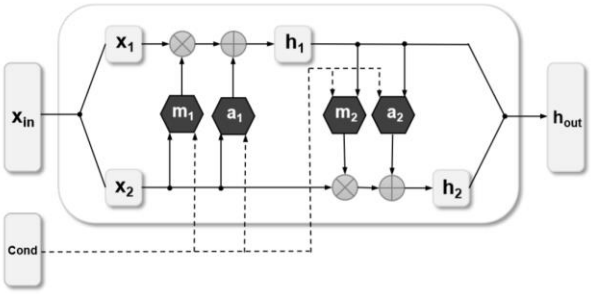

Figure 2. Pipeline of conditional affine coupling block(CACB)

III. PRELIMINARY

Estimating the underlying distribution density $\tilde{p}(x)$ of the dataset X is a classical challenging task in machine learning. To learn the most representative generative model, the KL divergence between real distribution $\tilde{p}(x)$ and estimation distribution $\mathrm{q}(x)$ has to be minimized

$$\begin{aligned}\mathrm{KL}\big(\tilde{p}(x) \parallel q(x)\big) &= \int \tilde{p}(x) \log \frac{\tilde{p}(x)}{q(x)} dx \\ &= E_{x\sim\tilde{p}(x)}\left[\log \frac{\tilde{p}(x)}{q(x)}\right] \\ &= c - E_{x\sim\tilde{p}(x)} \log q(x) \end{aligned} \quad (1)$$

or equally, maximizing the likelihood function $E_{x\sim\tilde{p}(x)} \log q(x)$. The basic idea of the modern generative model is to introducing a latent variable z and then convert $\mathrm{q}(x)$ into an integral formula of the following distribution

$$\mathrm{q}(x) = \int q\,(x,z)dz = \int q(z)q(x|z)dz \quad (2)$$

where the prior distribution $q(z)$ of the latent variable z can be set as a common distribution density, such as the standard Gaussian density. The conditional distribution $q(x|z)$ presents a generative procedure, which can be conditional Gaussian density or Fermi-Dirac density.

However, the integral Formula (2) is intractable to optimization. Instead of minimizing $\mathrm{KL}\big(\tilde{p}(x) \parallel q(x)\big)$, VAE introduce a posterior distribution $\mathrm{p}(z|x)$, referred to as the encoder procedure, and descend to minimize the KL divergence of the joint distribution density

$$\begin{aligned}&\mathrm{KL}\big(p(x,z) \parallel q(x,z)\big) \\ &= \mathrm{KL}\big(\tilde{p}(x)p(z|x) \parallel q(z)q(x|z)\big)\end{aligned}$$

$$= \iint \tilde{p}(x)p(z|x)\log\frac{\tilde{p}(x)p(z|x)}{q(x|z)q(x)}dzdx$$
$$= E_{x\sim\tilde{p}(x)}\left[\int p(z|x)\log\frac{p(z|x)}{q(x|z)q(z)}dz\right]$$
$$= E_{x\sim\tilde{p}(x)}\left[E_{x\sim p(z|x)}[-\log q(x|z)] + KL\left(p(z|x) \parallel q(z)\right)\right] \quad (3)$$

which is an upper bound of $\mathrm{KL}\left(\tilde{p}(x) \parallel q(x)\right)$ and is usually easy to calculate. The first item of Formula (3) is the reconstruction loss, and the second item is the KL loss of VAE. One of the problems of VAE is that the generated images are usually blurry. Owing to the Gaussian assumption of $\mathrm{p}(z|x)$ and the upper bound of optimization, the representation ability of VAE is restricted. Moreover, [11] indicate that the generated image features of VAE are unimodal, which means that VAE cannot capture the low-distribution modes of the real probability distribution density.

The normalizing flow-based invertible generative models take a different way, which supposes the conditional distribution $\mathrm{q}(x|z)$ as a Fermi-Dirac density

$$\mathrm{q}(x|z) = \delta\left(x - G^{-1}(z)\right) \quad (4)$$

and tackle the aforementioned integral Formula (1) directly by a well-designed G(z), which needs to ensure not only the invertibility

$$\mathrm{x} = G^{-1}(z) \Leftrightarrow z = \mathrm{G}(x) \quad (5)$$

but also, the tractable computability of Jacobian determinant

$$\frac{\partial \mathrm{G}(x)}{\partial x} \quad (6)$$

If we set the prior distribution $\mathrm{q}(z)$ as a standard multivariate Gaussian density

$$\mathrm{q}(z) = \frac{1}{(2\pi)^{D/2}} exp\left(-\frac{1}{2}\|z\|^2\right) \quad (7)$$

estimation distribution $\mathrm{q}(x)$ can be inferred by the integral transformation under the assumption of what G(z) is invertible

$$\mathrm{q}(x) = \mathrm{q}\left(G(x)\right) = \frac{1}{(2\pi)^{D/2}} exp\left(-\frac{1}{2}\|G(x)\|^2\right)\frac{\partial \mathrm{G}(x)}{\partial x} \quad (8)$$

whose logarithmic form is

$$\log \mathrm{q}(x) = -\frac{D}{2}\log(2\pi) - \frac{1}{2}\|\mathrm{G}(x)\|^2 + \log\frac{\partial \mathrm{G}(x)}{\partial x} \quad (9)$$

which is the objective function of the normalizing flow-based invertible generative model. The invertibility is to satisfy the generative procedure, and the tractable computability of the Jacobian determinant is to facilitate the calculation of the loss function. To meet the requirements, the strategy of the normalizing flow-based invertible generative model is to use affine coupling blocks to construct $\mathrm{G}(x)$. The method is presented in detail in the following section.

IV. APPROACHES

Our goal is to directly synthesize samples of unseen classes by explicitly modeling the underlying distribution of training data by using a powerful CNFG model. We can then train an ordinary supervised learning classifier by using synthesized unseen data and real seen data. The classifier can be any off-the-peg model, such as support vector machine (SVM) and SoftMax classifier. The overall pipeline of our model is illustrated in Figure 1.

TABLE I. STATISTICS OF FIVE BENCHMARKS

Dataset	Total class	Seen class	Unseen class	Total instance	Train instance	Test instance (unseen/seen)	Attributes
AwA1	50	40	10	30475	19832	5685/4958	85
AwA2	50	40	10	37332	23527	7913/5882	85
CUB	200	150	50	11788	7057	2679/1764	312
SUN	717	645	72	14340	10320	1440/2580	102
aPY	32	20	12	15339	5932	7924/1483	64

A. Affine coupling block

The affine coupling block is the basic module of CNFG model, which was proposed by NICE [17] and popularized by Glow [18]. It is a combination of additive coupling block and multiplicative coupling block. An affine coupling block first splits the input x_{in} into x_1 and x_2, and then transforms $[x_1, x_2]$ into $[h_1, h_2]$ by applying the affine coupling transformation

$$h_1 = x_1$$
$$h_2 = x_2 \otimes exp\left(m_2(x_1)\right) + a_2(x_1) \quad (10)$$

whose inverse is

$$x_1 = h_1$$
$$x_2 = \left(h_2 - a_2(h_1)\right) \oslash exp\left(m_2(x_1)\right) \quad (11)$$

and the lower triangular Jacobians matrix is

$$\frac{\partial f}{\partial x} = \begin{pmatrix} \mathbb{I}, & \mathbb{O} \\ \mathrm{em}\otimes\frac{\partial m_2}{\partial x_1}\otimes x_2 + \frac{\partial a_2}{\partial x_1}, & em \end{pmatrix} \quad (12)$$

where $\mathrm{em} = \exp\left(m_2(x_1)\right)$.

To improve the nonlinearity of transformation, [19] extends the affine coupling block by introducing a more complex affine transformation

$$h_1 = x_1 \otimes exp\left(m_1(x_2)\right) + a_1(x_2)$$
$$h_2 = x_2 \otimes exp\left(m_2(h_1)\right) + a_2(\mathrm{h}_1) \quad (13)$$

The conditional variant of the affine coupling block was first proposed by cINN [20]. Since the sub-transformation (m_i and a_i) of each affine coupling block is not inverted, cINN concatenate the conditional information c to the input of the sub-transformation and does not violate the assumption of invertibility. We can obtain the conditional affine coupling transformation by simply replacing $m_i(x)$ and $a_i(x)$ with $m_i(x, c)$ and $a_i(x, c)$ in Formula (13), respectively. The pipeline of the conditional affine coupling block is presented in Figure 2.

B. Architecture of our model

Our model is built on the principle of the conditional affine coupling transformation. The overall pipeline of the proposed model is presented in detail in Figure 1. Specifically, several

TABLE II. RESULTS IN THE GENERALIZED ZSR SETTING

Method	SUN			CUB			AWA1			AWA2			aPY		
	acc^{s}_{avg}	acc^{us}_{avg}	H	acc^{s}_{avg}	acc^{us}_{avg}	H	acc^{s}_{avg}	acc^{us}_{avg}	H	acc^{s}_{avg}	acc^{us}_{avg}	H	acc^{s}_{avg}	acc^{us}_{avg}	H
CONSE	6.8	39.9	11.6	1.6	**72.2**	3.1	0.4	88.6	0.8	0.5	**90.6**	1.0	0.0	**91.2**	0.0
CMT	8.1	21.8	11.8	7.2	49.8	12.6	0.9	87.6	1.8	0.5	90.0	1.0	1.4	85.2	2.8
DAP	4.2	25.1	7.2	1.7	67.9	3.3	0.0	**88.7**	0.0	0.0	84.7	0.0	4.8	78.3	9.0
IAP	1.0	37.8	1.8	0.2	72.8	0.4	2.1	78.2	4.1	0.9	87.6	1.8	5.7	65.6	10.4
SSE	2.1	36.4	4.0	8.5	46.9	14.4	7.0	80.5	12.9	8.1	82.5	14.8	0.2	78.9	0.4
DEVISE	16.9	27.4	20.9	23.8	53.0	32.8	13.4	68.7	22.4	17.1	74.7	27.8	4.9	76.9	9.2
SJE	14.7	30.5	19.8	23.5	59.2	33.6	11.3	74.6	19.6	8.0	73.9	14.4	3.7	55.7	6.9
LATEM	14.7	28.8	19.5	15.2	57.3	24.0	7.3	71.7	13.3	11.5	77.3	20.0	0.1	73.0	0.2
ALE	21.8	33.1	26.3	23.7	62.8	34.4	16.8	76.1	27.5	14.0	81.8	23.9	4.6	73.7	8.7
SAE	8.8	18.0	11.8	7.8	54.0	13.6	1.8	77.1	3.5	1.1	82.2	2.2	0.4	80.9	0.9
SYNC	7.9	43.3	13.4	11.5	70.9	19.8	8.9	87.3	16.2	10.0	90.5	18.0	7.4	66.3	13.3
ESZSL	11.0	27.9	15.8	12.6	63.8	21.0	6.6	75.6	12.1	5.9	77.8	11.0	2.4	70.1	4.6
GFZSL	0.0	39.6	0.0	0.0	45.7	0.0	1.8	80.3	3.5	2.5	80.1	4.8	0.0	83.3	0.0
CVAE	-	-	26.7	-	-	34.5	-	-	47.2	-	-	51.2	-	-	-
SE-GZSL	30.5	40.9	34.9	53.3	41.5	46.7	67.8	56.3	61.5	68.1	58.3	62.8	-	-	-
GMMN	37.7	39.7	38.7	55.9	49.1	52.3	**70.1**	51.5	59.3	77.3	46.3	57.9	64.4	28.5	39.5
GDAN	**89.9**	38.1	**53.4**	**66.7**	39.3	49.5	-	-	-	67.5	32.1	43.5	75.0	30.4	**43.4**
CNFG (ours)	41.2	**43.6**	42.3	62.3	47.1	**53.6**	69.5	57.4	**62.8**	**69.3**	58.1	**63.2**	**66.8**	31.0	42.3

conditional affine coupling blocks can be cascaded and constructed into a more complex and powerful generative model, referred to as conditional normalizing flow-based generative model (CNFG). Suppose each affine coupling transformation denoted as f_i, where $\mathrm{i} = 1,2,\dots,\mathrm{n}$, then we can obtain a composite function

$$\begin{aligned} \mathrm{z} &= f_n\left(h^{(n)}, c\right) \\ &= f_n\left(f_{n-1}\left(h^{(n-1)}, c\right)\right) \\ &= \cdots \\ &= f_n\left(f_{n-1}(\dots f_0(x,c)\dots)\right) \\ &= G(x,c) \end{aligned} \tag{14}$$

which is the conditional variant of the well-desired $G(x,c)$ in Formula (4). We generally denote the encoding procedure of the CNFG model as $G(x,c;\theta)$. The inverse or decoding procedure of the network is denoted as $G^{-1}(z,c;\theta)$, representing the generative procedure. Our goal is to optimize the network parameters θ by maximizing the logarithmic form of q(x) in Formula (9). The Jacobian matrix of affine coupling transformation strictly adheres to the lower or upper triangular form, as seen in Formula (12) and we can view it as a constant

$$\frac{\partial f}{\partial x} = \mathrm{c} \tag{15}$$

The partial derivative with respect to x of $\mathrm{G}(x,c)$ is also a constant C.

$$\frac{\partial \mathrm{G}(x,c)}{\partial x} = \frac{\partial f_n}{\partial f_{n-1}} \cdot \frac{\partial f_{n-1}}{\partial f_{n-2}} \dots = \frac{\partial f_0}{\partial x} = C \tag{16}$$

Thus, the objective function $\log \mathrm{q}(x,c)$ in Formula (9) can be simplified as

$$\log \mathrm{q}(x,c) = -\frac{1}{2}\|G(x,c)\|^2 + \mathrm{C} \tag{17}$$

The maximum log likelihood $E_{x\sim\tilde{p}(x)} \log q(x,c)$ is equal to the minimum of $1/2\,\|\mathrm{G}(x,c)\|^2$. Finally, the maximum likelihood training procedure can be implemented by simply minimizing the mean square value of $\mathrm{z} = G(x,c)$.

Encoder procedure: We first extract all image features from real image instances by using the pre-trained ResNet101, then designate the 2048-dim image features x of the seen class and the corresponding attributes vector *c* as the inputs of the CNFG model. The input image feature x is transformed into the latent variable z via 12 conditional affine coupling blocks. The optimization objective is to minimize the mean square of z.

Decoder procedure: Once the generative model is learned, we can sample the latent variable z from the multivariate Gaussian density and then combine z with the attribute vector c of the unseen/seen class as the inputs of the generative model. The output is the synthesized instance of the corresponding class. We can generate any number of instances for the unseen/seen class because the data distribution of the latent variable z and the attribute vector A of the seen/unseen class is already known.

Classification procedure: With the "pseudo" annotated data, an off-the-peg supervised classification model can be trained. In the proposed model, we use the SVM as the final classifier. In the generalized setting, the classifier is biased toward the seen class to a certain extent if we only use the original annotated instance of the seen class and the synthesized instance of the unseen class. Thus, we augment the original annotated instance of the seen class with the synthesized instance of seen class as well.

V. EXPERIMENT

We present experiments on the five publicly released benchmarks: Animals with Attributes 1 (**AWA1**) [21] , Animals with Attributes 2 (**AWA2**) [22], Caltech-UCSD Birds 200 (**CUB**) [23], SUN attribute database (**SUN**) [24] and aPascal & aAyahoo (**aPY**) [25]. The statistical data for these benchmarks is listed in Table I. In accordance with [22], we use the harmonic mean average accuracy of the seen and unseen class in the testing set as the evaluation metric, which is defined as

$$\mathrm{H} = 2 * \frac{acc^{seen}_{avg} * acc^{unseen}_{avg}}{acc^{seen}_{avg} + acc^{unseen}_{avg}} \tag{18}$$

where acc^{seen}_{avg} and acc^{unseen}_{avg} represent the average accuracy of the seen and unseen class, respectively. We realize our model

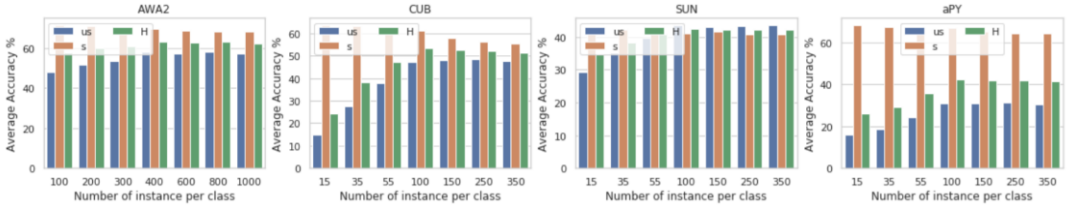

Figure 3. Analysis of the effects of the number of synthesized instances on acc_{avg}^{unseen}(**us**), acc_{avg}^{seen}(**s**) and **H** scores.

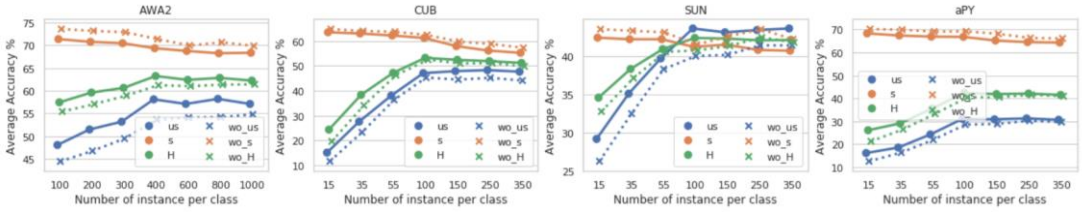

Figure 4. Analysis of the effects of the data augmentation on acc_{avg}^{unseen}(**us**), acc_{avg}^{seen}(**s**), **H** scores, where **wo_*** represent without data augmentation

with the deep learning framework Keras. Specifically, we construct CNFG model with 12 affine couple blocks. Each affine couple block consists of the *Shuffle*, *Split*, *Concat*, *AffineCouple* and *Subnetworks* modules. The *Shuffle* module first disrupts the order of the input vector for fully mixing the information and increasing the nonlinearity of transformation. The *Split* and *Concat* modules are responsible for dividing the input x into two parts before affine transformation and reassemble it back after affine transformation. The *AffineCouple* module is implemented with the corresponding subnetworks a_i and m_i by using a multilayer perceptron with 3 or 5 hidden layers and receiving the attribute vector directly as conditional information. The hidden layers have 1,024 units that are half the dimension of the image features. All multilayer perceptron subnetworks use ReLU activation function and appropriate dropout layers to avoid over-fitting. In the training procedure, we use Adam as our optimizer with the hyper-parameters learning rate = 0.001 and momentum = (0.9, 0.999).

A. Results Analysis

In accordance with [22], we randomly divide all seen class instances into 80% and 20% parts in the class level for the generalized setting. The two parts are denoted as X_{seen}^{train} seen and X_{seen}^{test}, respectively. We train our generative model on the training set D_{seen}^{train} , which consists of X_{seen}^{train} and the corresponding attributes representation A_{seen}^{train}, then synthesize the pseudo instance of the seen and unseen classes by using our trained generative model and denoting them as X_{seen}^{pseudo} and X_{unseen}^{pseudo} , respectively. We finally combine these pseudo instances with the original seen data X_{seen}^{train} to fit a multi-class linear SVM as the final classifier. Once the final classifier is fitted, we evaluate the performance of the fitted classifier on X_{seen}^{test} and X_{unseen}^{test} using average accuracy metric and denoting them as acc_{avg}^{seen} and acc_{avg}^{unseen}. Naturally, we calculate the harmonic mean value by using Formula (18) and present all acc_{avg}^{seen}, acc_{avg}^{unseen} and H scores on each dataset, as seen in Tables II.

Table II shows that the family of mapping-based methods have pervasive higher acc_{avg}^{unseen} scores and lower acc_{avg}^{seen} , H scores. These results demonstrate that the bias problem prevails in mapping-based ZSR methods, and those methods are not suitable for the generalized ZSR setting. Meanwhile, the family of ZSR methods based on the generative model made a good tradeoff between acc_{avg}^{seen} and acc_{avg}^{unseen} . The proposed CNFG model improves over the mapping-based method by 25% on AWA1/AWA2 benchmark and achieves the significant performance on the other benchmarks. We attribute this improvement to the efficiency of the generative model at capturing the underlying distributions. We also compare the proposed model with recent state-of-the-art methods based on the generative models. As shown in the bottom area of Table II, the proposed method outperforms most of the ZSR methods that are based on VAE or GAN. This difference in performance is attributed to the following: 1) the VAE and GAN have their own inner limitations, which are stated in Section III. 2) the proposed models can capture some low-distribution modes of real data density by virtue of its stable and exact log-likelihood maximum training procedure.

B. Number of Synthesized Instances

Although we can synthesize any number of instances for each class by using the generative model, it is inadvisable to arbitrarily generate large amounts of synthesized instances. In this section, we conduct several control experiments to evaluate the effects of NUM on the final classifier, which denotes the number of synthesized instances per class. We generate 7 different numbers of synthesized instances for each class by using the trained CNFG model. Specifically, we generate [15, 35, 55, 100, 150, 250, 350] instances per class for the CUB, SUN, and aPY datasets, as well as [100, 200, 300, 400, 600, 800, 1000] instances per class for the AWA1/AWA2 dataset, which is the large-scale dataset on the basis of the number of instances per class. The result is shown in Figure 3. Observation based on Figure 3 include the following: 1) With an increase in the

number of synthesized instances, the average accuracy of the unseen class acc_{avg}^{unseen} improves significantly for all datasets. This increasement is expected because there are no instances of the unseen class exist in the beginning. By contrast, this evident improvement demonstrates that the synthesized unseen data are very close to the real testing data of the unseen class. Thus, this increasement also indirectly proves the generative ability of the proposed model. 2) As the number of synthesized instances increases, the average accuracy of the seen class acc_{avg}^{seen} mildly decreases in the testing set. This result is expected because the final classifier is trained on an increasing number of synthesized unseen data. The higher acc_{avg}^{seen} may be irregular in the beginning, given that the final classifier is unintentionally biased toward the seen class. 3) The harmonic mean score H first increases rapidly but does not improve substantially upon reaching certain level because the acc_{avg}^{seen} score and acc_{avg}^{unseen} scores change in opposite directions with an increase in the number of synthesized instances. Thus, synthesizing numerous instances is unnecessary.

C. Data Augmentation

In the generalized setting, we can fit the final classifier by merely using synthesized unseen data and original seen data or augment original seen data with synthesized seen data. To evaluate the effectiveness of data augmentation, we trained two different models for each benchmark with or without augmented seen data in the generalized setting. As shown in Figure 4, the accuracy of the seen class acc_{avg}^{seen} is apparently higher than that of the unseen class acc_{avg}^{unseen} if we train the final classifier only on original seen data and synthesized unseen data. This mean that the final classifier is biased toward the seen class to a certain extent. The gap between the acc_{avg}^{seen} and acc_{avg}^{unseen} scores has been alleviated in the case of data augmentation. The improvement is explained by the fact that the synthesized seen data enlarging the decision space of SVM not only for the seen class but also for the unseen class.

REFERENCES

[1] Miloš Radovanovic, Alexandros Nanopoulos, and Mirjana Ivanovi ´ c. Hubs in space: Popular nearest neighbors in high-dimensional data. Journal of Machine Learning Research, 11(Sep):2487–2531, 2010.

[2] Angeliki Lazaridou, Georgiana Dinu, and Marco Baroni. Hubness and pollution: Delving into cross-space mapping for zero-shot learning. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 270–280, 2015.

[3] Yanwei Fu, Timothy M Hospedales, Tao Xiang, and Shaogang Gong. Transductive multi-view zero-shot learning. IEEE transactions on pattern analysis and machine intelligence, 37(11):2332–2345, 2015.

[4] Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In European Conference on Computer Vision, pages 52–68. Springer, 2016.

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

[6] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In International Conference on Machine Learning, pages 2152–2161, 2015.

[7] Elyor Kodirov, Tao Xiang, and Shaogang Gong. Semantic autoencoder for zero-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3174–3183, 2017.

[8] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2021–2030, 2017.

[9] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6857–6866, 2018.

[10] Maxime Bucher, Stéphane Herbin, and Frédéric Jurie. Generating visual representations for zero-shot classification. In Proceedings of the IEEE International Conference on Computer Vision, pages 2666–2673, 2017.

[11] Ashish Mishra, Shiva Krishna Reddy, Anurag Mittal, and Hema A Murthy. A generative model for zero shot learning using conditional variational autoencoders. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 2188–2196, 2018.

[12] Vinay Kumar Verma, Gundeep Arora, Ashish Mishra, and Piyush Rai. Generalized zero-shot learning via synthesized examples. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4281–4289, 2018.

[13] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5542–5551, 2018.

[14] Rafael Felix, Vijay BG Kumar, Ian Reid, and Gustavo Carneiro. Multimodal cycle-consistent generalized zero-shot learning. In Proceedings of the European Conference on Computer Vision (ECCV), pages 21–37, 2018.

[15] He Huang, Changhu Wang, Philip S Yu, and Chang-Dong Wang. Generative dual adversarial network for generalized zero-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 801–810, 2019.

[16] Mert Bulent Sariyildiz and Ramazan Gokberk Cinbis. Gradient matching generative networks for zero-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2168–2178, 2019.

[17] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. arXiv preprint arXiv:1410.8516, 2014.

[18] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In Advances in Neural Information Processing Systems, pages 10215–10224, 2018.

[19] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. arXiv preprint arXiv:1605.08803, 2016.

[20] Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Guided image generation with conditional invertible neural networks. arXiv preprint arXiv:1907.02392, 2019.

[21] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pages 951–958. IEEE, 2009.

[22] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning-the good, the bad and the ugly. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4582–4591, 2017.

[23] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010.

[24] Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In 2012 IEEE Conference on Computer Vision and Pattern Recognition, pages 2751–2758. IEEE, 2012.

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