Transfer Increment for Generalized Zero-Shot Learning

Liangjun Feng and Chunhui Zhao, *Senior Member, IEEE*

***Abstract*— Zero-shot learning (ZSL) is a successful paradigm for categorizing objects from the previously unseen classes. However, it suffers from severe performance degradation in the generalized ZSL (GZSL) setting, i.e., to recognize the test images that are from both seen and unseen classes. In this article, we present a simple but effective mechanism for GZSL and more open scenarios based on a transfer-increment strategy. On the one hand, a dual-knowledge-source-based generative model is constructed to tackle the missing data problem. Specifically, the local relational knowledge extracted from the label-embedding space and the global relational knowledge, which is the estimated data center in the feature-embedding space, are concurrently considered to synthesize the virtual exemplars. On the other hand, we further explore the training issue for the generative models under the GZSL setting. Two incremental training modes are designed to learn directly the unseen classes from the synthesized exemplars instead of the training classifiers with the seen and synthesized unseen exemplars together. It not only presents an effective unseen class learning but also requires less computing and storage resources in practical application. Comprehensive experiments are conducted based on five benchmark data sets. In comparison with the state-of-the-art methods, both the generating and training processes are considered for virtual exemplars by the proposed transfer-increment strategy, which results in a significant improvement in the conventional and GZSL tasks.**

***Index Terms*— Generalized zero-shot learning (GZSL), generative model, incremental learning, knowledge transfer, zero-shot learning (ZSL).**

I. Introduction

AS ONE of the most fundamental tasks in computer vision, object recognition attracted particular attention in recent years. Through learning from large-scale annotated data sets, intelligent perception models have obtained significant advances on the basic classification problem [1]–[3]. The supervised learning paradigm is driven by sufficient exemplars, abundant computing, and storage resources, which are usually fulfilled in well-designed circumstances. However, when it turns to the outdoor scenario and practical application, collecting and storing abundant training images for rare objects can be difficult or sometimes just impossible.

To recognize objects without training exemplars, zero-shot learning (ZSL) [4], [5] and generalized ZSL (GZSL) [6], [7] are explored in succession. In ZSL and GZSL, the attributes, textual description, and word vector of the class labels are often used as the auxiliary information to bridge the gap between the seen and unseen classes [8]–[13]. This technique makes it possible to identify a new object by just having a description of it. However, since the work proposed in the seminal paper of Lampert *et al.* [4], [5], ZSL has been studied in the unrealistic setting where the test data are assumed to come from unseen classes by default. Because the seen objects are always more common in the real world, the classification accuracy for the seen classes may be as important as that of the unseen ones [6]. The unreasonable restriction of ZSL is lifted in GZSL, and both seen and unseen classes are allowed at the test phase, which significantly improves the practicability.

Although published methods [14]–[18] have presented considerable promise on the conventional ZSL task, empirical studies conducted by Chao *et al.* [6] and Xian *et al.* [7] reported that conventional ZSL approaches suffered from severe performance degradation in the GZSL scenario. The classical direct attribute prediction (DAP) method [4] cannot overcome the inherent shift problem, in which the discriminant functions are often biased to the seen classes. Rahman *et al.* [19] produced one principal direction for each seen class. These directions were combined to obtain the principal direction for each unseen class, which conducted a unified approach for conventional, generalized, and few-shot learning. Feng *et al.* [20] addressed GZSL as a triple verification problem and proposed a unified optimization of regression and compatibility functions. Wang *et al.* [21] developed an asymmetric graph-based model to preserve simultaneously the class-level semantic manifold and the instance-level visual manifold in a latent space for GZSL. Yu *et al.* [22] proposed the latent-space-encoding method, which performs the interactions of different modalities by a feature-aware latent space in an implicit way. In addition, some generative models are also applicable [23]–[28] for GZSL. In general, these methods learn a probability distribution for each seen class using the information extracted from the embedding spaces and extrapolate it to synthesize the distributions of the unseen classes. Wang *et al.* [24] assumed that the local relational knowledge (LRK) learned from the label-embedding space can be transferred to synthesize the virtual centers of the unseen classes. However, the feature-embedding space that

Manuscript received August 6, 2019; revised January 17, 2020 and April 22, 2020; accepted June 28, 2020. Date of publication July 14, 2020; date of current version June 2, 2021. This work was supported in part by the NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization under Grant U1709211 and in part by the Zhejiang Key Research and Development Project under Grant 2019C01048 and Grant 2019C03100. *(Corresponding author: Chunhui Zhao.)*

The authors are with the State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: liangjunfeng@zju.edu.cn; chhzhao@zju.edu.cn).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2020.3006322

is the target domain is not considered at all when the LRK is extracted. Directly transferring knowledge from the label-embedding space to the feature-embedding space is rough, and it is better to calibrate the LRK based on the knowledge of the feature-embedding space.

The generative models synthesize exemplars and change the GZSL task into a conventional supervised recognition problem. However, after the generating process, few articles present an effective strategy to deal with the synthesized exemplars and the seen exemplars for the GZSL task. Guo *et al.* [25], Wang *et al.* [26], and Zhao *et al.* [27] trained classifiers with the synthesized exemplars and focused on the conventional ZSL task, which neglected the GZSL problem. Jurie *et al.* [23] and Verma *et al.* [28] combined a certain number of synthesized exemplars and seen exemplars together to train the attribute classifiers or the soft-max regressors for the recognition of both seen and unseen classes in the GZSL setting. Here, the mentioned conventional training style is termed the combination learning mode (CLM). Because GZSL requires the recognition ability for both seen and unseen classes, whenever the demand for learning new unseen classes is proposed, not only the synthesized exemplars but also the data of the seen classes are required by the CLM. This makes the CLM inflexible and inefficient for the learning of the unseen classes. It is also unreasonable to assume that the numerous seen exemplars are always available during the learning of the unseen classes in the wild or some outdoor scenes. Improved training modes are desired to learn from the synthesized exemplars effectively and balance the accuracy between the seen and unseen classes well for the GZSL task.

In this article, a transfer-increment strategy is proposed to improve both the generating and training processes of virtual exemplars for GZSL. Our approach consists of transfer and increments two stages. In the transfer stage, two knowledge sources are used to generalize learning from the seen to unseen classes. First, the LRK that denotes the sparse mapping from the seen classes to the unseen classes in label embedding is considered. The definition of LRK will be further clarified in the related work. Second, it is observed that not only the data in each class form a tight cluster but also the overall data set does. This can be seen from the visualization shown in Fig. 1, where the benchmark data set is projected onto the 3-D space. This characteristic is used to estimate the center of the unseen classes in the feature-embedding space as a supervisor to mitigate the bias between the synthesized virtual exemplars and the real unseen exemplars. Here, the knowledge estimated is termed the global relational knowledge (GRK). Both the local and GRK are used to construct the generative model. The proposed transfer technique is linear and easy to implement. In the increment stage, we further explore the training issue for the generative models in the GZSL setting. Incremental learning is first adopted as an effective strategy to learn directly from the virtual exemplars for the unseen classes and remain the ability of recognizing the seen classes. Two concise linear incremental algorithms are embedded into the classical probabilistic model to implement the category increment. Both the cases where the data of the seen classes can be used or not during the incremental learning stage are considered, which means we design the incremental learning with the data of the seen class mode (IWM) and the incremental learning without the data of the seen class mode (IOM) simultaneously to provide a more general approach for practical application.

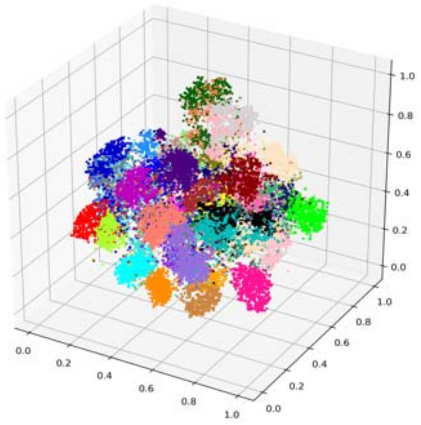

Fig. 1. Visualization of all classes in the AWA benchmark data set [4] (convolutional neural network (CNN) features) in the 3-D space by t-distributed stochastic neighbor embedding (t-SNE) [29]. Different kinds of exemplars are scattered with different colors. The features used here are extracted by GoogleNet [30], which is pretrained on the ImageNet data set without additional adjustments.

The contributions are summarized as follows.

1) A dual-knowledge-source-based generative model is proposed using the defined GRK and the classic LRK to achieve a simple but effective knowledge transfer.
2) Incremental training modes are first designed to learn directly the unseen classes from the synthesized exemplars and obtain a more effective training process for the GZSL task.

The remainder of this article is organized as follows. In Section II, we briefly review the related work. Then, the transfer-increment learning strategy is presented, followed by the case studies on five benchmark data sets. Finally, the conclusions are drawn.

II. Related Work

A. Embedding Technique

Feature and label embeddings are the foundational techniques in the task of recognizing the unknown objects, which transform the images and labels from their original spaces into the designed feature- and label-embedding spaces [4], [5], [31]. The illustration of the embedding technique is shown in Fig. 2. In the feature-embedding space, the features are often extracted by the network for deep representations. In the label-embedding space, each class of the data set is described by its attributes or word vector, and so on [8]–[13]. Here, the animals with attributes (AWA) data set [4] is taken as an example for a better understanding of the attribute description. The "blue whale" class in the data set can be described by many attributes, including "limbs (false)," "blue (true)," "sea (true)," and "huge (true)." Similarly, the "chimpanzee" can be described by the same attributes, including "limbs (true)," "blue (false)," "sea (false)," and "huge (false)." The attribute descriptions can be processed to attribute vectors by the one-hot encoding technique and used in the label-embedding space as the fine-grained class-level representations of different classes.

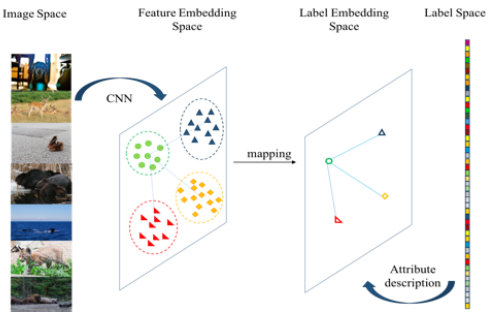

Fig. 2. Illustration of feature and label embeddings in ZSL. The original images are extracted as deep features by the convolutional network, and the class labels are described by the attribute vectors. The mapping learned in ZSL is usually between the feature-embedding space and the label-embedding space.

The application of embedding layers for ZSL can be divided into two main paradigms [7], [32], [33]. The first compatibility-based method learns the linear or nonlinear cross-modal mapping with discriminative losses and returns a compatibility score to decide the class of the unseen exemplar. For example, square loss and implicit regularization were applied to the ranking formulation by Romera-Paredes and Torr [34]. Neural networks were also introduced to learn the nonlinear mapping by Socher *et al.* [35]. Schonfeld *et al.* [36] used the aligned variational autoencoders to learn an additional latent embedding between the feature- and label-embedding spaces and, subsequently, trained a classifier on the sampled latent features of the seen and unseen classes. Ji *et al.* [37] learned a hashing model by using the label-embedding space to transfer knowledge from the seen class to the unseen class and implement the cross-model ZSL. The second probability-based method can be represented by the classical DAP and indirect attribute prediction (IAP) methods [4], [5], which learn a number of probabilistic learners (classifiers or regressors) and combine the scores to make predictions.

B. Relational Knowledge Transfer

The LRK that denotes the linear mapping between the seen and unseen classes in the feature and label embeddings was formally defined by Wang *et al.* [26] in the LRK transfer (LRKT) method. Wang proved that the learners for the seen classes could be directly used for the unseen classes when the LRK is equal in the two embedding spaces and extended it to generate virtual unseen exemplars. Zhao *et al.* [27] empirically validated that LRK worked well under the ZSL setting and extended the work of Wang *et al.* [26] to the transductive setting. In addition, we note that Fu *et al.* [38] made a similar assumption called connectivity assumption before Wang and Zhao. Fu thought that if the projection of a test image and an unseen class prototype were associated with the strongly related seen class prototypes, they should be close. LRK learns the mapping between the seen and unseen classes in label embedding and has been applied to mitigate the domain-shift problem between the seen and unseen classes.

C. Generative Models

The generative models tackle the essential problem of ZSL that there are no exemplars of unseen classes available for model training and, hence, can be popular in recent years. Besides the mentioned relational knowledge-transfer method [26], Verma *et al.* [28] designed a feedback mechanism and used the off-the-shelf classifier at the test stage to present an improved performance. Yang *et al.* [39] used the diffusion-regularization method to synthesize the unseen visual data. The well-known generative adversarial network (GAN) [23], [40]–[44] is also applied in the ZSL field. Jurie *et al.* [23] compared the performances of four different network-based generative models on ZSL, including the conditional GAN, generative moment matching network, and denoising autoencoder. Xian *et al.* [41] proposed a new GAN that synthesized features conditioned on the class-level semantic information. The model offered a shortcut directly from a semantic descriptor of a class to a class-conditional feature distribution. Zhu *et al.* [42] used the GAN that took the noisy text descriptions about an unseen class as input and the generated synthesized visual features for this class. These generative methods presented promising performance on the ZSL and GZSL tasks. However, most of them are complex-network-based generative methods. The model implementations are actually quite difficult in practical, and the adversarial training can also be time-consuming and unstable. Simpler methods may be desired for a more efficient generation process.

D. Incremental Learning

Incremental strategy presents an efficient paradigm for the learning tasks. Due to the missing data problem, only a few articles mentioned the strategy in the ZSL field. Nan *et al.* [45] adopted the linear-discriminant analysis and QR decomposition to realize incremental ZSL based on the IAP method. When more real exemplars (classes) are selected, the model can be incrementally updated to learn the information about the attributes for better predictions. Because the IAP has biases to the seen classes under the GZSL setting [6], this method is actually limited to the ZSL task. Similarly, based on the IAP model, Kawewong [46] used the unsupervised incremental neural networks to learn more kinds of attributes for the ZSL task, which aimed at the scenario that the attributes were labeled gradually from different users in an online incremental manner. Ferreira *et al.* [47] developed an online adaptive strategy for spoken language understanding, which was actually an application of ZSL. Both the experiments and data sets are different from the standard ZSL. It should be noted that none of the mentioned methods apply the incremental learning technique to generative models to learn the unseen classes and report the results under the standard GZSL setting [6], [7]. Differently, two incremental training modes are developed for the proposed generative model in this article, and comprehensive results based on five benchmark data sets are reported.

III. Methodology

In this section, we first present the basic idea and notations. The proposed transfer-increment strategy is described in Sections III-A and III-B, followed by the complexity analysis of the three training modes. Finally, we conclude the overall pipeline to show the application of our method.

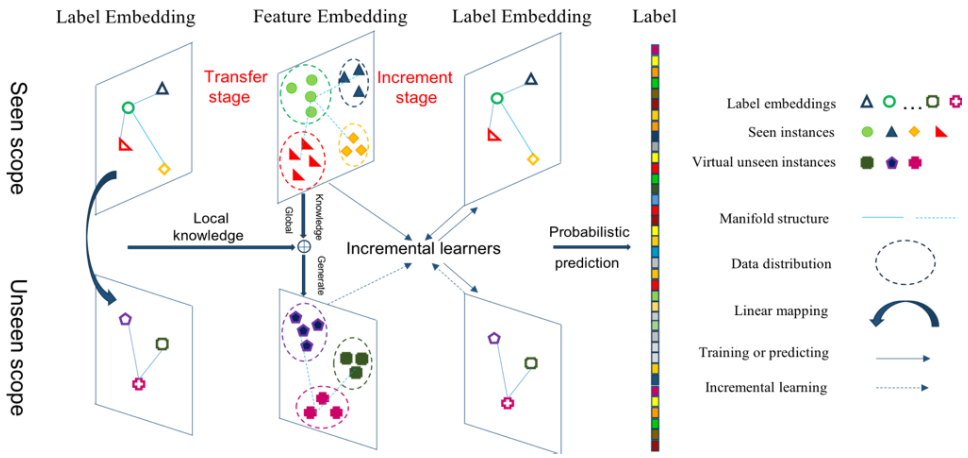

Fig. 3. Illustration of the proposed transfer-increment strategy, which consists of two stages. In the transfer stage, the LRK of label embedding and the GRK of the seen features are used together to construct the generative model and synthesize virtual exemplars for the unseen classes. In the increment stage, the incremental learners that have been trained on the seen exemplars keep learning from the generated exemplars for the unseen classes. Both the seen and unseen classes are recognized by the classical DAP model.

A. Statement and Notations

The proposed approach is designed in a transfer-increment framework, which is shown in Fig. 3. In the transfer stage, both the local knowledge and global knowledge are extracted for unseen classes to construct the generative model. On the one hand, the LRK learns the relationship between the classes from the class prototypes in label embedding. On the other hand, the feature center of the seen classes is estimated as the GRK to offer the feature-related information. The concurrent consideration for the local and global knowledge sources aims to synthesize the reliable exemplars for the unseen classes. As for the increment stage, the incremental learning is adopted to offer an efficient training for the unseen classes and reduce the dependence of the model on the data of the seen classes. Both the final categories of the seen and unseen exemplars are determined based on the DAP method. The designed transfer-increment strategy focuses on the generating and training processes of the virtual exemplars to obtain an improved performance for the generative models under the GZSL setting.

Provided that a data set $\Im$ with N exemplars and D dimension is given as $\Im = \{\boldsymbol{X}, \boldsymbol{Y}\}$, where $\boldsymbol{X} \in \mathbb{R}^{D\times N}$ is the data matrix and $\boldsymbol{Y} \in \mathbb{R}^{1\times N}$ is the label vector. In the ZSL and GZSL settings, the item of $\boldsymbol{Y}$ is discrete, which makes it a classification problem. The set of all classes in $\Im$ can be denoted as $T = \{1, \ldots, p, p+q, \ldots, p+q\}$, where p is the number of seen classes and q is the number of unseen classes. Specifically, the set of seen classes is denoted as $S = \{1, \ldots, p\}$, the set of unseen classes is denoted as $U = \{p+1, \ldots, p+q\}$, and $S \cap U = \varnothing$. The data set is divided into two scopes by U and S, namely, seen scope and unseen scope. Therefore, $\boldsymbol{X} = \{\boldsymbol{X}_S, \boldsymbol{X}_U\}$ can be obtained, where $\boldsymbol{X}_S$ denotes the data of the seen classes and $\boldsymbol{X}_U$ denotes the data of the unseen classes. In addition, we have $\boldsymbol{Y} = \{\boldsymbol{Y}_S, \boldsymbol{Y}_U\}$. Both $\boldsymbol{X}_U$ and $\boldsymbol{Y}_U$ are unavailable during the training stage. It is worth mentioning that the attribute information $\boldsymbol{A} \in \mathbb{R}^{\mathrm{AD}\times(p+q)}$ is offered in ZSL, where AD denotes the dimension of the attribute information. Similarly, $\boldsymbol{A}$ can be divided into $\boldsymbol{A}_S \in \mathbb{R}^{\mathrm{AD}\times p}$ and $\boldsymbol{A}_U \in \mathbb{R}^{\mathrm{AD}\times q}$ in the isolated scopes. After introducing the attribute information $\boldsymbol{A}$ for all the classes, each class is described by an AD-dimension attribute vector, and the label $\boldsymbol{Y} = \{\boldsymbol{Y}_S, \boldsymbol{Y}_U\} \in \mathbb{R}^{1\times N}$ can be extended to the attribute label $\boldsymbol{Z} = \{\boldsymbol{Z}_S, \boldsymbol{Z}_U\} \in \mathbb{R}^{\mathrm{AD}\times N}$ by the merging of $\boldsymbol{A}$ and $\boldsymbol{Y}$. The conventional ZSL aims to achieve the learning from S to U based on the set $\{\boldsymbol{X}_S, \boldsymbol{Y}_S, \boldsymbol{Z}_S, \boldsymbol{A}\}$. GZSL attempts to implement the learning from S to T using the same set.

B. Dual-Knowledge-Source-Based Transfer Stage

In this section, the transfer stage is introduced. We use two kinds of knowledge, i.e., LRK and GRK, to construct the generative model, which are extracted from the label-embedding space and the feature-embedding space, respectively.

The data logic of GZSL is shown in Fig. 4(a), in which $\boldsymbol{E}_S \in \mathbb{R}^{D\times p}$ is the mean matrix of the seen class data and each column $\boldsymbol{e}_i \in \mathbb{R}^{D\times 1}$ in $\boldsymbol{E}_S$ denotes the mean vector of the ith seen class. In addition, the basic ideas of DAP [4], [5], LRKT [25], and our generative model are comparatively presented in Fig. 4(b)–(d). LRKT transfers and applies the LRK to synthesize the virtual mean matrix of the unseen classes $\hat{\boldsymbol{E}}_U \in \mathbb{R}^{D\times q}$ and tackle the shift problem of DAP. However, it considers nothing about the real exemplars during extracting LRK. Differently, our method estimates the center of the unseen classes in the feature-embedding space as the GRK to make the overall distribution of the synthesized unseen classes closer to the real one. GRK is defined based on the following assumption.

Assumption: In the ZSL and GZSL settings, the overall center of the data set could be approximated by the seen classes and transferred to that of the unseen classes. This is formulated as

$$\boldsymbol{e}_c \approx \mathrm{mean}(\boldsymbol{E}_S) = \frac{1}{p}\sum_{i=1}^{p}\boldsymbol{e}_i$$
$$\Rightarrow \frac{1}{q}\sum_{i=p+1}^{p+q}\boldsymbol{e}_i = \mathrm{mean}(\boldsymbol{E}_U) \approx \boldsymbol{e}_c \quad (1)$$

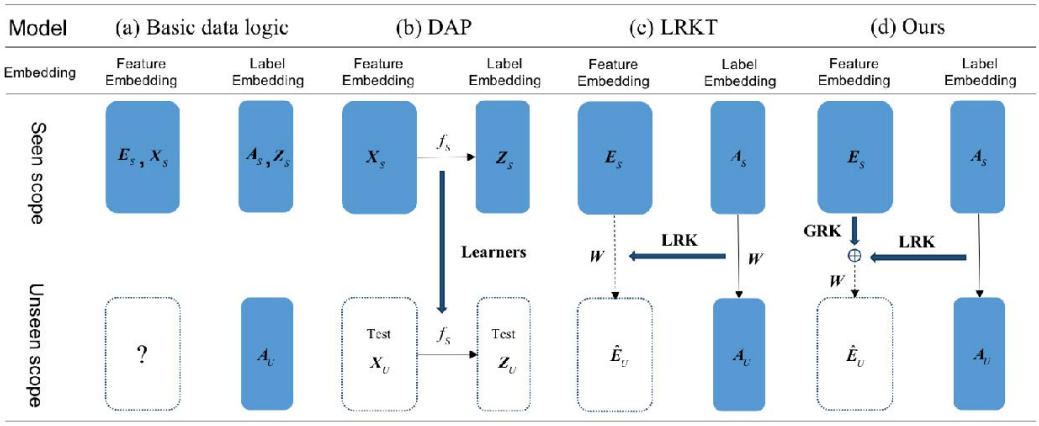

Fig. 4. Comparison of different methods under the given notations, including (a) basic data logic, (b) DAP method, (c) LRKT method, and (d) proposed transfer model. The blue arrow denotes the knowledge transfer, the solid arrow denotes the mapping of a certain function, and the dashed arrow denotes the generating process of the unseen classes. $\boldsymbol{E}_S$ denotes the mean matrix of all seen classes and $\hat{\boldsymbol{E}}_U$ denotes the generated mean matrix of the unseen classes.

where $\boldsymbol{e}_c$ is the overall center of the data set and $\boldsymbol{E}_U \in \mathbb{R}^{D\times q}$ is the real mean matrix of the unseen classes.

The approximation in (1) converges when the number of seen and unseen classes, i.e., p and q, approaches infinite and the number of exemplars in each class is identical, which can be proved as below.

Proof: The distribution of the ith class in the data set $\Im$ is denoted as $\varphi(\boldsymbol{e}_i)$, where $\boldsymbol{e}_i$ is the center of φ. Based on the central limit theorem [48], [49], we have

$$\lim_{p+q\to\infty}\sum_{i=1}^{p+q}\varphi(\boldsymbol{e}_i) \to G(\boldsymbol{e}_c) \tag{2}$$

where G denotes the Gaussian distribution Equation (2) means the overall distribution of $\Im$ tends to the Gaussian distribution with the increasing number of classes.

Considering that the data set is divided by the seen and unseen scopes, the formulation could be rewritten as

$$\lim_{p+q\to\infty}\left(\sum_{i=1}^{p}\varphi(\boldsymbol{e}_i)+\sum_{i=p+1}^{p+q}\varphi(\boldsymbol{e}_i)\right) \to G(\boldsymbol{e}_c). \tag{3}$$

Both the seen and unseen classes are the subsets of the data set. According to the Glivenko–Cantelli theorem [50], [51], if the subsets are randomly selected, with the increasing p and q, we have

$$\lim_{p\to\infty}\sum_{i=1}^{p}\varphi(\boldsymbol{e}_i) = \lim_{q\to\infty}\sum_{i=p+1}^{p+q}\varphi(\boldsymbol{e}_i) \to G(\boldsymbol{e}_c). \tag{4}$$

From (4), it is observed that the distributions of the seen and unseen classes tend to the same overall distribution $G(\boldsymbol{e}_c)$ of $\Im$ and share the same data center $\boldsymbol{e}_c$, which is denoted as

$$\begin{aligned}\lim_{p\to\infty}\frac{1}{N_S}\sum_{i=1}^{p}\sum_{k=1}^{n_i}\boldsymbol{x}_k^i &= \lim_{q\to\infty}\frac{1}{N_U}\sum_{i=p+1}^{p+q}\sum_{k=1}^{n_i}\boldsymbol{x}_k^i \\ &\to \lim_{p+q\to\infty}\frac{1}{N}\sum_{i=1}^{p+q}\sum_{k=1}^{n_i}\boldsymbol{x}_k^i\end{aligned} \tag{5}$$

where n_i is the number of exemplars in the ith class, $\boldsymbol{x}_k^i$ is the kth exemplar in the ith class, N_S and N_U are the number of exemplars of the seen and unseen classes, and $N = N_S + N_U$. With the identical n_i, the assumption in (1) can be obtained as follows:

$$\lim_{p\to\infty}\frac{1}{p}\sum_{i=1}^{p}\boldsymbol{e}_i = \lim_{q\to\infty}\frac{1}{q}\sum_{i=p+1}^{p+q}\boldsymbol{e}_i \to \boldsymbol{e}_c. \tag{6}$$

Based on the introduced assumption for GRK, our model aims to synthesize directly the virtual centers $\hat{\boldsymbol{E}}_U$ for the unseen classes using $\boldsymbol{E}_S$, which can be formulated as $\hat{\boldsymbol{E}}_U = \kappa(\boldsymbol{E}_S)$, where κ denotes the mapping between the seen and unseen two scopes. Without loss of generality, κ is set as a linear mapping $\boldsymbol{W} \in \mathbb{R}^{p\times q}$ in the feature-embedding space. The seen and unseen classes are represented by $\{\boldsymbol{E}_S, \hat{\boldsymbol{E}}_U = \boldsymbol{E}_S\boldsymbol{W}\}$ and $\{\boldsymbol{A}_S, \boldsymbol{A}_U\}$ in the feature- and label-embedding spaces, respectively. We transfer LRK and GRK for $\boldsymbol{W}$ simultaneously by the defined loss function L as follows:

$$\begin{aligned}L(\boldsymbol{W}|\boldsymbol{A}_S, \boldsymbol{A}_U, \boldsymbol{E}_S, \boldsymbol{e}_c) = \|\boldsymbol{A}_S\boldsymbol{W} - \boldsymbol{A}_U\|_2^2 + \alpha\|\text{mean}(\boldsymbol{E}_S\boldsymbol{W}) \\ - \boldsymbol{e}_c\|_2^2 + \beta\|\boldsymbol{W}\|_1.\end{aligned} \tag{7}$$

There are three items in L. The first item, i.e., $\|\boldsymbol{A}_S\boldsymbol{W} - \boldsymbol{A}_U\|_2^2$, is related to LRK [25], which denotes that the projection of label embedding is approximated to that of the cluster centers in the feature-embedding space. The second item, i.e., $\|\text{mean}(\boldsymbol{E}_S\boldsymbol{W}) - \boldsymbol{e}_c\|_2^2$, is for GRK, which requires that the center of the synthesized $\hat{\boldsymbol{E}}_U = \boldsymbol{E}_S\boldsymbol{W}$ should be close to $\boldsymbol{e}_c$ as much as possible. $\boldsymbol{e}_c$ here is the center of the unseen classes transferred from that of the seen classes. Finally, the l_1-norm is adopted to make $\boldsymbol{W}$ sparse and avoid overfitting. α and β are the weight parameters for the GRK and regularization item, respectively. Though minimizing the loss function in (7), the model aims to learn the relationship between the classes in the label-embedding space and the overall center of the real exemplars in the feature-embedding space simultaneously. While the single object of LRKT neglects the bias between the real and virtual exemplars, the concurrent consideration for GRK aims to make the center of the synthesized unseen

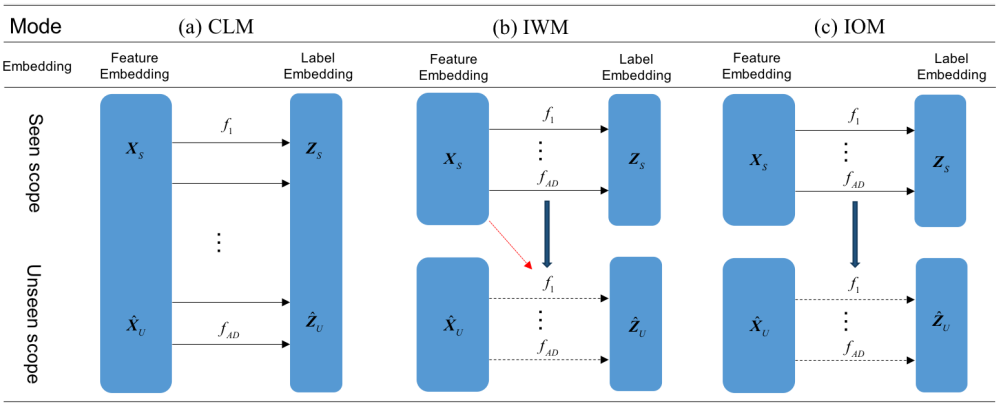

Fig. 5. Comparison of (a) CLM, (b) IWM, and (c) IOM under the GZSL setting. f denotes the learner (classifier or regressor), the solid arrow denotes the general training process, the black dashed arrow denotes the incremental learning process, and the red arrow denotes that $\boldsymbol{X}_S$ is used during incremental learning.

classes closer to that of the real exemplars. In addition, for the solution, the objective function is convex and can be efficiently solved by the convex optimization toolbox.

After obtaining the transfer mapping $\boldsymbol{W}$, it is ready to generate labeled virtual exemplars for the unseen classes by assigning the distribution $\varphi(\boldsymbol{e}_i)$ of each class to the Gaussian distribution $G(\boldsymbol{e}_i, \Phi_i)$. The covariance matrix Φ_i is regularized as a diagonal variance matrix $\Phi_i = \sigma \boldsymbol{I}$, where σ is a predefined prior knowledge. In addition, the mean vector $\boldsymbol{e}_i$ is generated by

$$\boldsymbol{e}_i = \boldsymbol{E}_S \boldsymbol{W}_i \tag{8}$$

where $\boldsymbol{W}_i$ is the ith column of $\boldsymbol{W}$.

In addition, an attractive characteristic of our generative model in (7) and (8) is that all the inputs for synthesizing the unseen exemplars, i.e., $\{\boldsymbol{A}_S, \boldsymbol{A}_U, \boldsymbol{E}_S, \boldsymbol{e}_c\}$, are of class level instead of sample level, which means only a small amount of data need to be stored for the generating process. This property can be integrated with the followed incremental training modes to present greater practicability in the outdoor scenarios.

C. Incremental Training Stage

The virtual labeled data $\{\hat{\boldsymbol{X}}_U, \hat{\boldsymbol{Y}}_U, \hat{\boldsymbol{Z}}_U\}$ can be generated for the unseen classes at the transfer stage, where $\hat{\boldsymbol{X}}_U \in \mathbb{R}^{D\times M}$ is the synthesized unseen exemplars, $\hat{\boldsymbol{Y}}_U \in \mathbb{R}^{1\times M}$ is the synthesized unseen label, $\hat{\boldsymbol{Z}}_U \in \mathbb{R}^{\mathrm{AD}\times M}$ is the synthesized unseen attribute label, and M is the number of synthesized exemplars. To learn efficiently from the virtual exemplars and reduce the dependence of the transfer-increment strategy on the data of the seen classes, two incremental training modes, i.e., IWM and IOM, are designed.

In comparison with the conventional CLM, the basic ideas of IWM and IOM are presented in Fig. 5. The conventional CLM combines $\hat{\boldsymbol{X}}_U$ and $\boldsymbol{X}_S$ together to form the training data $\hat{\boldsymbol{X}} = [\boldsymbol{X}_S \mid \hat{\boldsymbol{X}}_U]$ and learn the attribute label $\hat{\boldsymbol{Z}} = [\boldsymbol{Z}_S \mid \hat{\boldsymbol{Z}}_U]$, where $\boldsymbol{Z}_S \in \mathbb{R}^{\mathrm{AD}\times N_S}$ is the attribute label of the seen exemplars and N_S is the number of seen exemplars. CLM is the basic DAP method. Comparatively, IWM and IOM initialize the attribute learners using the seen data $\{\boldsymbol{X}_S, \boldsymbol{Z}_S\}$ and incrementally learn the synthesized exemplars $\{\hat{\boldsymbol{X}}_U, \hat{\boldsymbol{Z}}_U\}$. The exemplars of the seen classes $\boldsymbol{X}_S$ are used during the incremental learning process of IWM. This has shown benefits in keeping the recognition ability for the seen classes. As for IOM, this mode no longer requires the data of the seen classes during the learning for the unseen classes. Both the IWM and IOM conduct the final classification based on the classical DAP method. Obviously, the dependence of CLM, IWM, and IOM on the seen exemplars decreases in order.

1) Incremental Learning With the Data of Seen Classes Mode: IWM initializes attribute learners with the data of seen classes $\boldsymbol{X}_S \in \mathbb{R}^{D\times N_S}$ and the attribute label matrix $\boldsymbol{Z}_S = [z_1^S, \ldots, z_{\mathrm{AD}}^S]^T \in \mathbb{R}^{\mathrm{AD}\times N_S}$, where N_S is the number of seen exemplars. For an attribute label $z_i^S \in \mathbb{R}^{N_S\times 1}$ $(i = 1, \ldots, AD)$ of $\boldsymbol{X}_S$, a basic linear form of the attribute learner is presented with a regularization item as follows:

$$\min \left\| z_i^S - \boldsymbol{X}_S^T \boldsymbol{w} \right\|_{\beta_1}^{\alpha_1} + \gamma \left\| \boldsymbol{w} \right\|_{\beta_2}^{\alpha_2} \tag{9}$$

where $\alpha_1 > 0$, $\alpha_2 > 0$, β_1, and β_2 are typically kinds of norm regularization and $\boldsymbol{w} \in \mathbb{R}^{D\times 1}$ is the learned attribute learner. By taking $\alpha_1 = \alpha_2 = \beta_1 = \beta_2 = 2$, the object is set with the l_2-norm regularization, which is convex and has a better generalization performance. The value γ is the parameter of the regularization item. The object can be easily solved by the ridge regression theory

$$\boldsymbol{w} = \boldsymbol{X}_S^{T+} z_i^S \tag{10}$$

where $\boldsymbol{X}_S^{T+}$ is the pseudoinverse of $\boldsymbol{X}_S^T$ and is obtained by

$$\boldsymbol{X}_S^{T+} = \lim_{\gamma \to 0} \left(\gamma \boldsymbol{I} + \boldsymbol{X}_S \boldsymbol{X}_S^T\right)^{-1} \boldsymbol{X}_S. \tag{11}$$

Based on the solution, AD attribute learners can be trained for the seen classes using $\{\boldsymbol{X}_S, \boldsymbol{Z}_S\}$. Next, the same learners incrementally learn from the synthesized virtual data $\{\hat{\boldsymbol{X}}_U, \hat{\boldsymbol{Z}}_U\}$ for the unseen classes. This can be achieved by updating the pseudoinverse $\boldsymbol{X}_S^{T+}$ to $\hat{\boldsymbol{X}}^{T+}$

$$\hat{\boldsymbol{X}}^{T+} = [\boldsymbol{X}_S^{T+} - \boldsymbol{B}\boldsymbol{P}^T \mid \boldsymbol{B}] \tag{12}$$

where $\boldsymbol{P}^T = \hat{\boldsymbol{X}}_U^T \boldsymbol{X}_S^{T+}$

$$\boldsymbol{B} = \begin{cases} \boldsymbol{U}^+, & \text{if } \boldsymbol{U} \neq 0 \\ (\boldsymbol{I} + \boldsymbol{P}\boldsymbol{P}^T)\boldsymbol{X}_S^{T+}\boldsymbol{P}, & \text{if } \boldsymbol{U} = 0 \end{cases} \tag{13}$$

and

$$\boldsymbol{U} = \hat{\boldsymbol{X}}_U^{T+} - \boldsymbol{P}^T \boldsymbol{X}_S^T. \tag{14}$$

With the updated pseudoinverse, $\boldsymbol{w}$ can be updated to $\hat{\boldsymbol{w}}$ by

$$\hat{\boldsymbol{w}} = \hat{\boldsymbol{X}}^{T+} \hat{\boldsymbol{z}}_i \tag{15}$$

where $\hat{\boldsymbol{z}}_i = [z_i^{ST} | \hat{\boldsymbol{z}}_i^{UT}]^T \in \mathbb{R}^{(N_S+M)\times 1}$ and $\hat{\boldsymbol{z}}_i^U \in \mathbb{R}^{M\times 1}$ is the virtual attribute label in $\boldsymbol{Z}_U = [\hat{\boldsymbol{z}}_1^U, \ldots, \hat{\boldsymbol{z}}_{\mathrm{AD}}^U]^T \in \mathbb{R}^{\mathrm{AD}\times M}$.

IWM uses the incremental algorithm in (12)–(15) to update the attribute learners to learn the additional attribute knowledge of the unseen classes. During updating, the exemplars of the seen classes $\boldsymbol{X}_S$ are set as the intermediate information in (14) rather than the input data. The incremental algorithm allows IWM to learn the unseen classes and remain the knowledge for the seen classes without the entire retraining from the beginning.

2) Incremental Learning Without the Data of Seen Classes Mode: IOM learns all exemplars in an iterative manner, and the loss for each exemplar is minimized after iteration, which is similar with the adaptive strategy.

An exemplar pair is denoted as $(\boldsymbol{x}, z)$, where $\boldsymbol{x} \in \mathbb{R}^{D\times 1}$ is a feature vector and represents an exemplar, $z \in \mathbb{R}$ is a real value of a certain attribute label. A linear attribute mapping $\boldsymbol{w} \in \mathbb{R}^{D\times 1}$ for $(\boldsymbol{x}, z)$ is learned by minimizing the hinge loss, which is formulized as follows:

$$l_\varepsilon = \begin{cases} 0, & |z - \boldsymbol{x}^T \boldsymbol{w}| < \varepsilon \\ |z - \boldsymbol{x}^T \boldsymbol{w}| - \varepsilon, & \text{otherwise.} \end{cases} \tag{16}$$

Here, $\boldsymbol{w}$ is initialized randomly and updated by mining the loss l_ε for each exemplar pair $(\boldsymbol{x}, z)$. The objective function is given by

$$\begin{aligned} \boldsymbol{w}_{t+1} = & \underset{w}{\operatorname{argmin}} \frac{1}{2} \|\boldsymbol{w} - \boldsymbol{w}_t\|_2^2 + C\xi \\ & \text{s.t. } l_\varepsilon(\boldsymbol{w}; (\boldsymbol{x}_t, z_t)) \le \xi \text{ and } \xi \ge 0 \end{aligned} \tag{17}$$

where t denotes the tth iteration, ξ is a slack term, and C is the positive parameter of ξ. The object can be efficiently solved by the passive aggressive algorithm [52]–[54], and the solution is given by

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \operatorname{sign}(z_t - \hat{z}_t)\tau_t \boldsymbol{x}_t \tag{18}$$

where $\tau_t = \min\{C, l_t/\|\boldsymbol{x}_t\|^2\}$, $l_t = l_\varepsilon(\boldsymbol{w}; (\boldsymbol{x}_t, z_t))$, and $\hat{z}_t = \boldsymbol{x}_t^T \boldsymbol{w}_t$.

Similarly, based on the model in (16)–(18), AD attribute learners can be trained for the seen classes with $\{\boldsymbol{X}_S, \boldsymbol{Z}_S\}$. Since the iterative learning strategy learns the exemplars one by one, the data of the seen classes are not needed anymore after the initial training. The trained attribute learners can learn the unseen classes directly based on the synthesized exemplars. The iterative style learns the unseen classes much faster and presents a higher harmonic mean accuracy with a few virtual exemplars trained. It can be proved that although the solution of the passive aggressive algorithm of IOM for the unseen classes is without the exemplars of the seen classes, the knowledge loss of the seen classes is limited to achieve the recognition for both seen and unseen classes at the test stage [50].

D. Complexity Analysis

In this section, the time complexity of CLM, IWM, and IOM is analyzed. If the CLM model in (9) is retrained from the beginning for the unseen classes with both $\hat{\boldsymbol{X}}_U$ and $\boldsymbol{X}_S$, the complexity is $\vartheta_1(2(N_S+M)D^2+D^3)$. In addition, the complexity is $\vartheta_2(2N_S DM + 2MD^2 + D^3)$ for IWM to learn the unseen classes according to (12). The difference between ϑ_1 and ϑ_2 is calculated as follows:

$$\vartheta_1 - \vartheta_2 = 2N_S D(D - M). \tag{19}$$

In comparison with CLM, the incremental algorithm of IWM takes less time to learn the unseen classes when $D > M$. D is the dimension of features and M is the number of synthesized exemplars. In ZSL, the CNN features that are thousands of dimensions are usually used, so it is convenient for IWM to learn exemplars incrementally at small batches. As for IOM, the complexity of each iteration can be approximated as $\vartheta(2D)$. When M virtual exemplars are trained, the time complexity of IOM linearly increases as $\vartheta_3(2DM)$, which is more efficient than CLM.

Algorithm 1 Transfer-Increment Strategy for GZSL

Initial Training for Seen Classes

1 Learners $\{f_i, i = 1, \ldots, \mathrm{AD}\}$ is trained based on $\{\boldsymbol{X}_S, \boldsymbol{Z}_S\}$ using (9) or (17).

2 Extract side information $\{\boldsymbol{E}_S, \boldsymbol{e}_c\}$ of seen classes.

When *the demand for unseen classes U is proposed*

Transfer Learning for Virtual Exemplars

3 Obtain $\hat{\boldsymbol{E}}_U$ using (7) and (8).

4 Synthesize virtual exemplars $\{\hat{\boldsymbol{X}}_U, \hat{\boldsymbol{Z}}_U\}$ for unseen classes based on Gaussian distribution

Incremental Learning for Unseen Classes

5 Learners $\{f_i, i = 1, \ldots, \mathrm{AD}\}$ incrementally learn from $\{\hat{\boldsymbol{X}}_U, \hat{\boldsymbol{Z}}_U\}$ for the attribute knowledge of unseen classes

Test Phase at GZSL setting

6 Attribute values of $\{\boldsymbol{X}_U, \boldsymbol{X}_S^{\mathrm{test}}\}$ are predicted by learners, and exemplars are recognized based on DAP.

E. Overall Pipeline

The overall learning of the transfer-increment strategy is summarized in Algorithm 1, which consists of two main steps in practical application. At the initial learning stage, AD attribute learners $\{f_i, i = 1, \ldots, \mathrm{AD}\}$ are trained based on $\{\boldsymbol{X}_S, \boldsymbol{Z}_S\}$. The learners should be selected from (9) or (17), which depends on whether the data of the seen classes are available or not during the learning for the unseen classes. The side information of the seen classes $\{\boldsymbol{E}_S, \boldsymbol{e}_c\}$ should be extracted from the seen exemplars and reserved. At the end of this stage, the data of the seen classes can be abandoned according to the specific circumstance. The second stage is the learning for the unseen classes. Every time when the demand for learning the unseen classes is proposed, the transfer-increment strategy can be adopted. The generative model synthesizes the labeled virtual exemplars, based on which incremental algorithms can be applied to learn the unseen classes efficiently. Finally, the test exemplars are classified

TABLE I
STATISTICS OF FIVE BENCHMARK DATA SETS

No.	Dataset	Att.	Tot._Cla.	S_Cla.	U_Cla.	Tot._Ima.
1	CUB	312	200	150	50	11788
2	aPY	64	32	20	12	15339
3	SUN	102	717	645	72	14340
4	AWA	85	50	40	10	30475
5	AWA2	85	50	40	10	37322

Statistics in terms of the number of attributes, the number of classes, the number of seen classes, the number of unseen classes, and the number of images.

TABLE II
MODEL PARAMETERS FOR THE GZSL EXPERIMENT

No.	Dataset	α	β	σ	γ	C
1	CUB	0.200	2.000	0.200	0.010	0.001
2	aPY	0.100	1.000	0.300	0.010	0.001
3	SUN	0.200	1.000	0.200	0.010	0.001
4	AWA	0.500	2.000	0.400	0.010	0.010
5	AWA2	0.100	1.000	0.300	0.010	0.010

α, β, and σ are the parameters of the proposed generative model. γ and C are the parameters of the incremental learning stage.

based on the classical DAP method. The transfer-increment model (IOM) allows learning the unseen classes at multiple batches without the data of the seen classes, which requires for a little computing and storage resources in practical application.

IV. EXPERIMENTS

Experiments are conducted based on five benchmark data sets. The settings are detailed first, and then, the results of GZSL, the study on the effects of the proposed training modes, the validation for the proposed assumption, the parameter study, and the results of ZSL are reported in order.

A. Benchmark Data Sets and Experimental Settings

1) Benchmark Data Sets: The proposed approaches are evaluated on five benchmark data sets, including Caltech UCSD Birds (CUB) [55], aPascal and aYahoo (aPY) [13], SUN attributes (SUN) [56], AWA [4], and animals with attributes2 (AWA2) [7]. CUB is a data set with a few variations among different birds. aPY is a combined data set of aPY, in which aYahoo data set is collected from the Yahoo image search that is different from the ones in aPascal. SUN is a subset of the SUN scene data set with the fine-grained attributes. AWA and AWA2 are the standard data sets for ZSL. The statistics of these data sets are given in Table I. We use the standard seen/unseen split introduced by Xian *et al.* [7]. In the GZSL setting, we perform a 80%–20% split for each seen class exemplars; 20% exemplars of the seen classes are combined together with the unseen exemplars for test and the other 80% exemplars are used for training. In the ZSL setting, the test data only correspond to the unseen classes, and all the exemplars of the seen classes are set as the training data.

2) Feature and Label Embeddings: As repeatedly reported, the CNN features outperform the shallow features by a significant margin. Hence, we use ResNet101 [3] and InceptionV3 [1], which are pretrained on ImageNet to extract the visual features. We do not fine-tune the networks for any of the mentioned data sets. In specific, the last convolution layer of ResNet101 and the last pooling layer of InceptionV3 are used for representing the images. Both the dimensions of the extracted visual features are 2048. Since the images of AWA are not publicly available, only the features of GoogleNet [29] are used for the AWA data set, which are 1024 dimensions. As for label embedding, attributes and world vectors are commonly used in ZSL. Here, the continuous attributes are adopted for each data set, which have been reported to be more efficient than binary attributes.

3) Evaluation Metrics: The recognition performance is measured by top-1 accuracy [7], which is the percentage of the estimated labels (the ones with the highest scores) that match the correct labels. In the test of GZSL, we report the recognition accuracy of the unseen classes accU, the recognition accuracy of the seen classes accS, and the harmonic mean of the seen and unseen accuracies. The harmonic mean accuracy is given by

$$H = \frac{2 \times \text{accS} \times \text{accU}}{\text{accS} + \text{accU}}. \tag{20}$$

H is a good metric that estimates the biasness of any method toward seen classes. The harmonic mean will drop down significantly if a method is too biased to the seen or unseen classes.

4) Implementation Details: There are several parameters that should be determined by cross validation in the proposed transfer-increment strategy. For the five benchmark data sets, the weight parameters α and β in (7) are searched from 0.01 to 10. During generating exemplars for the unseen classes, the covariance matrices for the unseen classes are set as the diagonal variance matrix, and σ is searched from 0.1 to 1. The parameter of the regularization item for IWM γ in (9) is searched from 0.001 to 1, and the parameter of the slack term C for IOM in (17) is searched from 0.001 to 0.1. The searching results for the five data sets are shown in Table II. The effect of three main parameters α, β, and σ of our dual-knowledge-source-based generative model will be explored shortly in Section IV-E.

B. Results for GZSL Task

We compare our model with 16 state-of-the-art methods in the GZSL setting, including DAP [4], TVN [20], AGZSL [21], GVR [23], SYNC [24], LRKT [26], SE [28], ALE [32], DEVISE [33], ESZSL [34], CADA-VAE [36], f-CLSWGAN [41], LisGAN [42], CONSE [57], SAE [58], and CVAE [59]. Among them, DAP is the classical probabilistic model. ALE, DEVISE, ESZSL, CONSE, AGZSL, and SAE are the latent intermediate-embedding-space-based methods. TVN, GVR, SYNC, LRKT, SE, LisGAN, f-CLSWGAN, CADA-VAE, and CVAE are the generative models. The results of these methods in [7] and their articles are used for reference. Three kinds of training modes for generative model have been introduced, including CLM, IWM, and IOM. CLM is the conventional combination training mode. IWM and IOM are the proposed incremental training modes. Here, the virtual exemplars synthesized by our generative model are fit by three kinds of training modes, respectively. For a fair comparison,

TABLE III
COMPARISON IN GZSL SETTING

Method	CUB			aPY			SUN			AWA			AWA2		
	U	*S*	*H*	*U*	*S*	*H*	*U*	*S*	*H*	*U*	*S*	*H*	*U*	*S*	*H*
DAP(2013)	1.7	67.9	3.3	4.8	78.3	9.0	4.2	25.1	7.2	0.0	**88.7**	0.0	0.0	84.7	0.0
ALE(2013)	23.7	62.8	34.4	4.6	73.7	8.7	21.8	33.1	26.3	16.8	76.1	27.5	14.0	81.8	23.9
CONSE(2013)	1.6	**72.2**	3.1	0.0	**91.2**	0.0	6.8	**39.9**	11.6	0.4	88.6	0.8	0.5	**90.6**	1.0
DEVISE(2014)	23.8	53.0	32.8	4.9	76.9	9.2	16.9	27.4	20.9	13.4	68.7	22.4	17.1	74.7	27.8
ESZSL(2015)	12.6	63.8	21.0	2.4	70.1	4.6	11.0	27.9	15.8	6.6	75.6	12.1	5.9	77.8	11.0
SYNC(2016)	11.5	70.9	19.8	7.4	66.3	13.3	7.9	43.0	13.4	8.9	87.3	16.2	10.0	90.5	18.0
LRKT(2016)	16.9	41.5	24.0	12.8	48.2	20.3	17.8	23.7	20.3	15.4	83.8	26.1	17.3	88.7	28.9
GVR(2017)	26.8	72.0	39.1	–	–	–	–	–	–	32.3	81.3	46.2	–	–	–
SAE(2017)	7.8	54.0	13.6	1.1	82.2	2.2	8.8	18.0	11.8	1.8	77.1	3.5	1.1	82.2	2.2
CVAE(2018)	–	–	34.5	–	–	–	–	–	26.7	–	–	47.2	–	–	51.2
TVN(2018)	26.5	62.3	37.2	16.1	66.9	25.9	22.2	38.3	28.1	27.0	67.9	38.6	–	–	–
SE((2018)	41.5	53.3	46.7	–	–	–	40.9	30.5	34.9	56.3	67.8	61.5	58.3	68.1	62.8
fCLSWGAN*(18)	43.7	57.7	49.7	–	–	–	42.6	36.6	39.4	57.9	61.4	59.6	52.1	68.9	59.4
LisGAN(2019)	40.0’	60.0’	48.0’	–	–	–	–	–	–	–	–	–	–	–	–
CADA-VAE(19)	51.6	53.5	52.4	–	–	–	**47.2**	35.7	**40.6**	57.3	72.8	64.1	55.8	75.0	63.9
AGZSL(2019)	19.4	56.5	28.9	18.1	60.7	27.9	18.5	28.6	22.5	34.8	68.5	46.1	–	–	–
Ours+CLM+Res.	41.9	38.0	39.8	22.4	39.3	28.5	29.0	21.2	24.5	58.5	60.5	59.5	69.0	62.7	65.7
Ours+IWM+Res.	42.2	39.7	40.9	24.9	34.1	28.8	28.5	20.7	24.0	55.8	60.7	58.2	68.7	61.2	64.7
Ours+IOM+Res.	44.8	42.2	43.5	27.7	36.2	31.4	31.5	20.3	24.7	**61.5**	67.7	**64.4**	72.1	63.9	67.7
Ours+CLM+Doub.	45.2	48.1	46.5	26.3	31.8	28.8	39.3	20.9	27.3	–	–	–	78.1	60.4	68.1
Ours+IWM+Doub.	45.6	49.0	47.3	25.3	34.1	29.1	39.1	20.6	27.0	–	–	–	77.8	60.3	67.9
Ours+IOM+Doub.	**52.1**	53.3	**52.7**	**27.8**	38.7	**32.4**	32.3	24.6	27.9	–	–	–	**76.8**	66.9	**71.5**

U denotes the top-1 $accU$, *S* denotes the top-1 $accS$, and *H* denotes the harmonic mean. Res. denotes the features of ResNet101 are utilized, and Doub. denotes both of the features of ResNet101 and InceptionV3 are utilized. We measure top-1 accuracy in %. *Notes that f-CLSWGAN trains additional embedding models for testing, so its results may not be directly comparable with others. ’ Notes that the results of LisGAN are estimated from the seen-unseen curve in [40].

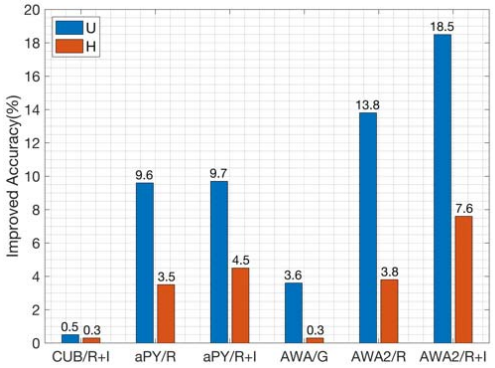

Fig. 6. Accuracy improvement over baseline. *U* denotes the improvement in accU and *H* denotes the improvement in the harmonic mean accuracy. R denotes the ResNet101 features, G denotes the GoogleNet features, and I denotes the InceptionV3 features. We measure top-1 accuracy in %.

the same ridge regression used by IWM is also set as the attribute learners for CLM.

The performance of the generative models is usually determined by the quality and quantity of the synthesized exemplars. For the quality, we have provided the model parameters in Table II. As for the quantity of the synthesized exemplars, the cross validation is conducted to find a suitable number for each data set and training modes. The effect of the number of generated exemplars will be explored shortly in the study of training modes. The comparison in the GZSL setting is shown in Table III, and the accuracy improvement over baselines is shown in Fig. 6.

First, for the results of the AWA and AWA2 data sets, the proposed transfer-increment strategy obtains the highest scores for the unseen classes and harmonic mean. Specifically, when only the ResNet101 features are used, the 72.1% unseen accuracy and 67.7% harmonic mean accuracy are obtained for AWA2. When both ResNet101 features and InceptiveV3 features are used, our transfer-increment strategy can perform even better, and the harmonic mean accuracy is above 70%, which is 7.6% higher than the baselines. Improvements on the scores of the unseen classes and harmonic mean for the CUB data set and the aPY data set are also observed. As for the SUN data set, the proposed method obtains higher scores than many benchmark methods. This data set has 14 340 fine-grained images, with attributes available at the image level (instead of class level). Because we combine the attributes of all images in a class to obtain the class-level attributes, the performance of our method for this data set may be affected.

Second, the results of CLM and IWM are observed to be similar to each other. Both of them are based on the direct attribute framework and use ridge regression as the attribute learners, while IWM adopts the incremental learning algorithm for unseen classes to avoid retraining from the beginning. For the five benchmark data sets, IOM usually presents the higher harmonic mean accuracy than CLM and IWM, and about 3%–6% improvement is observed.

Finally, the results of the proposed model in Table III are obviously biased to the scores of harmonic mean accuracy. The harmonic mean accuracy is a comprehensive metric which concurrently considers the performances on seen and unseen classes, and hence is widely used in GZSL task [7], [20], [21]. However, higher scores on the seen or unseen classes may be needed in specific cases, which can be achieved by adjusting the number of synthesized exemplars in the proposed model. Here, different number of exemplars of the unseen classes are generated to make the results, respectively, biased to the scores of the unseen classes, seen classes, and harmonic mean based on the IOM training mode. Both the features of

TABLE IV

RESULTS OF DIFFERENT BIASES TO SEEN, UNSEEN, AND HARMONIC MEAN ACCURACY

Dataset		CUB	aPY	SUN	AWA	AWA2
BU	*M*	1000	1000	5000	100	3000
	U	**54.1**	**32.5**	**34.1**	**68.6**	**77.2**
	S	35.3	16.2	21.2	38.4	43.8
	H	42.7	21.6	26.1	49.2	55.9
BS	*M*	3	9	19	3	13
	U	17.9	10.8	25.3	31.2	41.0
	S	**62.9**	**46.6**	**26.3**	**80.7**	**90.4**
	H	27.9	17.5	25.9	45.0	56.5
BH	*M*	39	25	600	10	70
	U	52.1	27.8	32.3	61.5	76.8
	S	53.3	38.7	24.6	67.7	66.9
	H	**52.7**	**32.4**	**27.9**	**64.4**	**71.5**

BU, BS, and BH denotes the results that are biased to unseen, seen, and harmonic mean accuracy, respectively. *M* denotes the number of generated exemplars for each unseen class. Both of the features of ResNet101 and InceptiobV3 are utilized here. We measure top-1 accuracy in %.

ResNet101 and InceptionV3 are used. The results are shown in Table IV. This property can be used to achieve different biases in practical. The setting for M is important for the model performance. In general, the more the virtual exemplars are generated, the stronger the recognition ability of the model for the unseen classes is. Hence, it is convenient to obtain the "BU" and "BS" values shown in Table IV. Three to twenty virtual exemplars make the model biased to the seen classes, and thousands of virtual exemplars make the model biased to the unseen classes. Here, a quadratic function is also given based on the statistics of the data sets to set intuitively a property value for M to obtain satisfactory harmonic mean accuracy "BH" shown in Table IV as follows:

$$M = 7.3574 \times 10^{-6} \times S \times N + 6.8982 \times 10^{-4} \times U \times N - 1.7344 \times 10^{-4} \times \mathrm{AD} \times N + 2.2996 \times 10^{-3} \times \mathrm{AD}^2 + 2.4061 \times 10^{-7} \times N^2 \quad (21)$$

where S denotes the number of seen classes, U denotes the number of unseen classes, N denotes the number of images, and AD denotes the number of attributes. Intuitively, larger M should be set for the data set that has more classes and images.

C. Training-Mode Evaluation

In this article, three kinds of training modes are introduced, including CLM, IWM, and IOM. Here, we explore the effects of different training modes on accS, accU, and harmonic mean accuracy when different number of virtual exemplars are used. In this experiment, ResNet101 features are used for the CUB data set, GoogleNet features are used for the AWA data set, and all parameters are set as those in Table II. The number of generated exemplars for each unseen class is adjusted from 1 to 1000 for the CUB data set, and 1–10 000 exemplars are generated for each unseen class of the AWA data set. The performance of the three training modes with varying number of virtual exemplars on the two benchmark data sets is shown in Fig. 7.

The results of the two benchmark data sets present similar trends. In the first two figures for the seen accuracy, accS of CLM and IWM decreases slowly with the increasing number of synthesized exemplars. In contrast, the accS of IOM decreases rapidly. From the middle two figures for the unseen accuracy, it is observed that accU of IOM increases much more quickly than that of CLM and IWM. Just a few exemplars can help IOM learn unseen classes well. Specifically, only ten virtual exemplars for each unseen class are needed for IOM to obtain 60% accU for the AWA data set, and 15 virtual exemplars are needed to obtain 40% accU for the CUB data set. However, to reach the same accuracy, CLM and IWM need about 7000 and 400 virtual exemplars for AWA and CUB, respectively. The last two figures show that only 10–40 exemplars are needed for IOM to obtain very high harmonic mean accuracy for the two benchmark data sets. As for CLM and IWM, the peaks of harmonic mean accuracy appear when 500 virtual exemplars are generated for the CUB data sets and 5000 virtual exemplars are generated for the AWA data sets.

TABLE V

COMPARISON OF TRAINING TIME FOR UNSEEN CLASSES

Time(s)	CUB	aPY	SUN	AWA	AWA2
CLM	291.41	52.09	122.00	174.67	254.13
IWM	148.01	39.71	84.89	110.01	133.71
IOM	13.57	2.35	3.69	4.08	6.61

The features of ResNet101 are used for CUB, aPY, SUN, and AWA2 dataset, and the GoogleNet features are used for AWA dataset. Twenty virtual exemplars are generated for each unseen class in experiment.

The training time for the unseen classes of the three training modes is compared in Table V. The training time of CLM, IWM, and IOM is decreased in order. Because the incremental learning style does not require for retraining from the beginning, IOM and IWM are faster than the conventional CLM.

In addition, IOM overcomes the dependence on the seen data during the learning for the unseen classes. Both the transfer and increment stages of our model do not use the seen exemplars, but some extracted class-level information and attribute learners, which saves the storage resources in practical. Here, we conduct a class-increment experiment, in which the unseen classes are leaned by IOM at ten batches. Except the initial training for the attribute learners, the seen exemplars are not used any more. The experiment is conducted on five data sets using the ResNet101 features (GoogleNet features for AWA). Fifty virtual exemplars are generated for each unseen class. The harmonic mean accuracy of multibatch learning is shown in Fig. 8. When a few unseen classes are added, IOM obtains a high score for harmonic mean accuracy. With the increase of unseen classes, the system still performs well.

Based on the above experiments, we summarize and compare the characteristics of the three training modes in Table VI. Both the CLM and IWM are required for the data of the seen classes during the learning of the unseen classes, while IOM is designed to learn the unseen classes without the seen data. This is an advantage for IOM in comparison with the CLM and IWM. From the complexity analysis in methodology and the results in Table V, it is observed that the training time of CLM, IWM, and IOM for the unseen classes is decreased in order. As for the accuracy, since CLM and IWM use the data of the seen classes, they can retain a good recognition

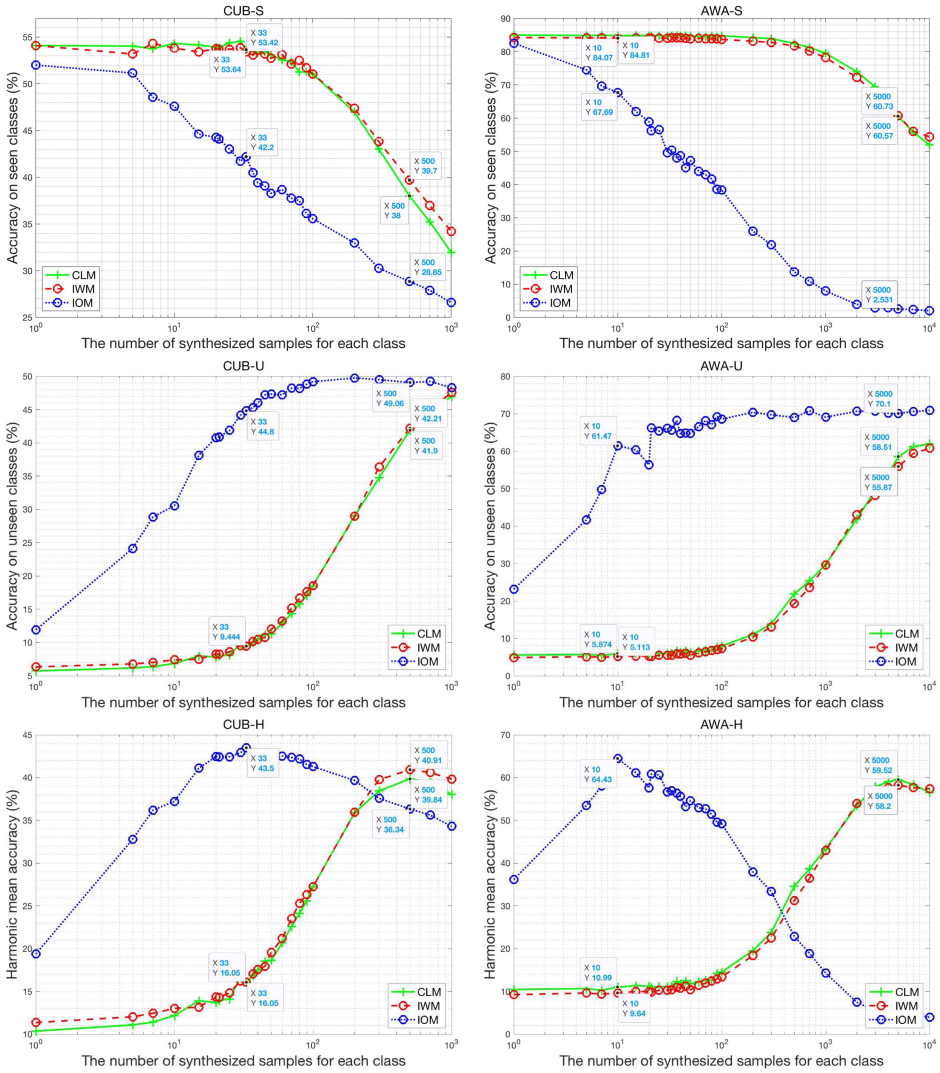

Fig. 7. Effects of the number of synthesized unseen exemplars on the performance of CLM, IWM, and IOM. U denotes the accU, S denotes the accS, and H denotes the harmonic mean accuracy. The features of ResNet101 are used for the CUB data set, and the features of GoogleNet are used for the AWA data set. The accuracy shown in text boxes corresponds to the results presented in Table III.

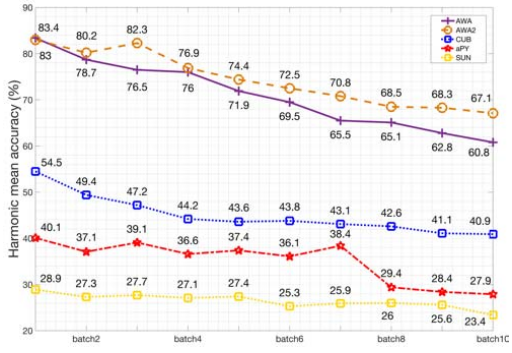

Fig. 8. Results of ten-batch learning by IOM for five benchmark data sets. More unseen classes are learned at each batch, and none of the seen exemplars are used during the incremental training.

TABLE VI
COMPARISON AMONG CLM, IWM, AND IOM

Model		CLM	IWM	IOM
Time complexity		ϑ_1	ϑ_2	ϑ_3
Training time for unseen classes		Long	Middle	**Short**
With a few exemplars generated for unseen classes	U	Low	Low	**High**
	S	High	High	**Middle**
	H	Middle	Middle	**High**
With numerous exemplars generated for unseen classes	U	**High**	**High**	High
	S	**Middle**	**Middle**	Low
	H	**High**	**High**	Middle

ϑ_1, ϑ_2, and ϑ_3 are presented in the complexity analysis subsection. U denotes the accuracy for unseen classes, S denotes the accuracy for seen classes, and H denotes the harmonic mean accuracy.

ability for the seen classes during the learning for the unseen classes. However, numerous virtual exemplars are needed for a high harmonic mean accuracy. In contrast, IOM usually obtains the highest harmonic mean accuracy with a few unseen exemplars generated. The iterative and adaptive learning strategy of the IOM learns unseen classes efficiently, which is validated by the results shown in Fig. 7. In addition, all the three modes can obtain satisfying results

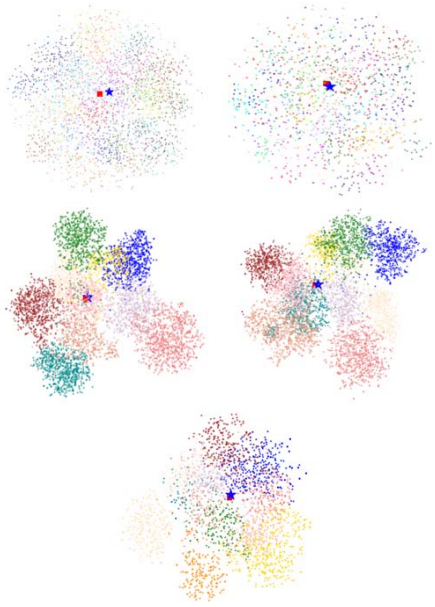

Fig. 9. Visualization of the unseen classes of five benchmark data sets in the 3-D space by t-SNE. The five figures correspond to CUB, SUN, AWA, AWA2, and aPY in order. Light-colored points represent the exemplars of the unseen classes. The red rectangle denotes the real center of the unseen classes and the blue star denotes the estimated center of the seen classes. The ResNet101 features and GoogleNet features are used.

when different number of exemplars are generated, which are shown in Table III.

D. Assumption Evaluation

In Section III, the center-related assumption is made to develop the knowledge-transfer model, and the distribution of each class is assigned to the Gaussian distribution.

The Gaussian distribution has been evaluated by Wang *et al.* [26] and Zhang and Saligrama [60] in the ZSL field. Here, we do not repeat the validation for the effectiveness of Gaussian distribution anymore. The center-related assumption assumes that unseen classes share the same data center with the seen classes based on the central limit theorem [48], [49] and Glivenko–Cantelli theorem [50], [51]. Here, we compare the real center of the unseen classes and the estimated center to evaluate the quality of assumption in (1). The features of the unseen classes of the five benchmark data sets (ResNet101 features of CUB, aPY, SUN, and AWA2, and GoogleNet features of AWA) are visualized in the 3-D space by t-SNE [29] and shown in Fig. 9. For the five data sets, the estimated centers by the seen exemplars and the real centers overlap significantly. By concurrently considering the estimated center as GRK, our dual-knowledge-source-based generative model is offered with the feature-related information and aims to make the center of the synthesized unseen exemplars closer to that of the real unseen exemplars.

For the effectiveness of LRK and GRK, it is validated by the comparison among the distribution of the real unseen classes, the distribution generated by LRK, and the distribution generated by the proposed model (the integration of LRK and GRK) on the AWA benchmark data set, which is shown in Fig. 10. The visualization is conducted based on t-SNE [29]. For the exemplars generated by LRK, the distributions of some unseen classes have already been quite similar to the real ones, especially in local relations. For example, the "red," "green," and "pink" classes are close to each other and similar to those in the real distributions. However, the single LRK actually does not consider the information in the feature-embedding space. The positions of some distributions may not be so perfect from the global view. After applying the GRK, the position-related information is offered in feature embedding, and the distributions of "blue," "yellow," "light pink," and "red" classes are slightly adjusted to better positions to make the whole distribution (center) of the unseen classes closer to the real ones. The distance between the real distributions (centers) and the generated distributions (centers) is comparatively shown in Table VII, which statistically validates the effectiveness of the proposed transfer model.

TABLE VII

DISTANCE COMPARISON BETWEEN THE REAL DISTRIBUTIONS AND GENERATED DISTRIBUTIONS ON THE AWA DATA SET

Distance		LRK	Ours(LRK+GRK)
Unseen classes	Class 1	28.22	25.98
	Class 2	24.27	21.62
	Class 3	20.71	19.17
	Class 4	9.45	7.86
	Class 5	11.45	11.07
	Class 6	18.27	17.57
	Class 7	12.82	11.40
	Class 8	35.89	31.09
	Class 9	19.70	18.39
	Class 10	18.04	14.43
	Center	**6.90**	**6.38**

Center denotes the overall center of unseen classes. The ordinary Euclidean distance is adopted as the metric here. The GoogleNet features (1024 dimensions) of AWA dataset are utilized.

E. Parameter Study

There are three important coefficients α, β, and σ in our dual-knowledge-source-based generative model. α is the coefficient of GRK, which denotes the weight of GRK in (7). β is the sparse coefficient, which denotes the weight of the sparse item in (7). σ is the given variance of the generative model. Model (7) is used to synthesize the virtual exemplars of the unseen classes. Here, the effects of the three coefficients on the accuracy of the unseen classes are studied under the GZSL setting. Two benchmark data sets, AWA and CUB, are used. We use the GoogleNet features for AWA and the ResNet101 features for CUB. IOM is adopted to learn the unseen classes. Twenty-five virtual exemplars for each unseen class are generated for both the data sets. During the cross validation of α, β and σ are set as 1 and 0.3, respectively. During the cross validation of β, α and σ are set as 0.1 and 0.3, respectively. During the cross validation of σ, α and β are set as 0.1 and 1, respectively. The results of cross validation are shown in Fig. 11.

The GRK coefficient α makes the center of the synthesized exemplars closer to the estimated center. For both data sets, a common trend is that the accuracy increases first and then decreases with the rising of α. The highest unseen accuracy can be obtained when α is 0.1–1. However, it is also noted that

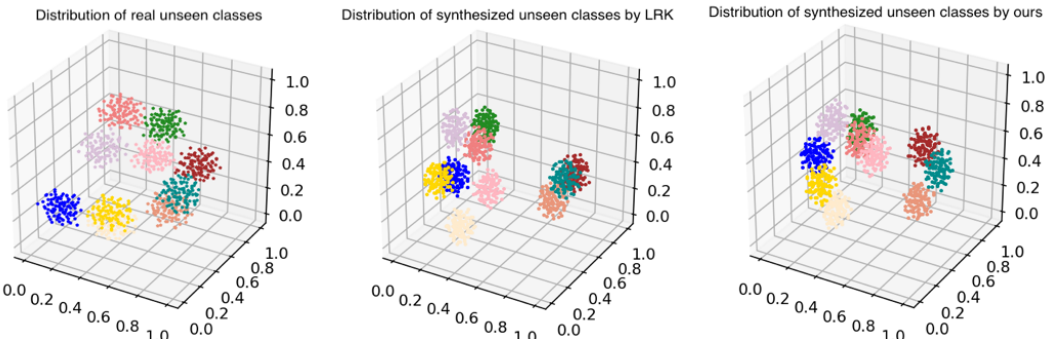

Fig. 10. Comparison among the distributions of the real unseen classes, the synthesized unseen classes by LRK [26], and the synthesized unseen classes by our transfer model (LRK + GRK) on the AWA benchmark data set. Fifty exemplars are generated for each unseen class by LRK and our model. For the distributions of the real unseen classes, the real mean value of each unseen class is used to generate 50 exemplars by the ways of LRK and our model for an intuitive presentation.

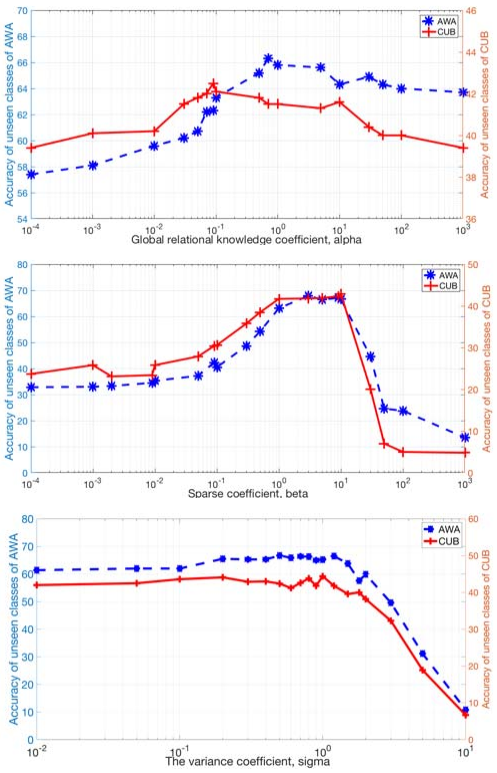

Fig. 11. Cross validation of α, β, and σ of the dual-knowledge-source-based generative model based on the AWA and CUB benchmark data sets (GoogleNet features for AWA and ResNet101 features for CUB). We report the accuracy in %.

adjusting α brings more accuracy improvement in the AWA data set. While the improvement in the CUB data set is about 3%, the improvement in the AWA data set is about 9%. This may be explained by the first figure in Fig. 9, in which the estimated center of the unseen classes of the CUB data set slightly deviates from the true center, but the estimated center of the unseen classes of the AWA data set is almost the same as the true center. As for the coefficients β and σ, the results show that when β is 1–10, the performance is the best, and the variance σ should not be too large.

F. Results for ZSL Task

For the conventional ZSL setting, we first generate virtual exemplars for the unseen classes, and the virtual exemplars are learned by the attribute learners for classification. Model (17) of IOM is set as the attribute learners. The parameters of the model are shown in Table VIII, and the number of synthesized exemplars for each class is determined by cross validation. Both ResNet101 and InceptionV3 features are used for the CUB, aPY, SUN, and AWA2 data sets. GoogleNet features are used for the AWA data set. The proposed model is compared with 13 state-of-the-art results, including ZSFGC [17], SCCT [18], SYNC [24], LRKT [26], ASTE [31], ALE [32], DEVISE [33], CMT [35], SAE [58], SJE [61], SSE [62], LATEM [63], and GFZSL [64], [65]. We present the results reported by Xian *et al.* [7] for reference. The comparison is shown in Table IX. Our method obtains the highest results on CUB, aPY, AWA, and AWA2 data sets. Only for the result of SUN, it is a little lower than a few state-of-the-art results. The SUN data set has 14 340 fine-grained images with attributes available at the image level instead of class level. Since we combine the attributes of all images in a class to obtain class-level attributes, the performance of our method for this data set may be affected.

In general, the proposed transfer-increment strategy can present advanced results in the conventional ZSL scenario. Occasionally, a few strong baseline methods perform better, such as the SE [28]. However, the motivations and advantages of the proposed transfer increment focus not only on the accuracy but also on the training efficiency and storage condition. As discussed in Section II-C, most of the popular generative models (including SE) are complex-network-based methods. The backpropagation and adversarial training require huge computing and storage resources. Instead, for the proposed

TABLE VIII
MODEL PARAMETERS FOR THE ZSL EXPERIMENT

Dataset	α	β	σ	C	M_1	M_2
CUB	0.1	1	1	0.001	50	40
aPY	0.1	1	1	0.001	30	20
SUN	0.1	1	0.3	0.001	3000	3000
AWA	1	1	1	0.01	400	–
AWA2	1	1	0.3	0.01	70	40

α, β, and σ are the parameters of the proposed generative model. C is the parameter of IOM. M_1 is the number of synthesized exemplars when ResNet101 features are utilized, and M_2 is the number of synthesized exemplars when both of ResNet101 features and InceptionV3 features are utilized.

TABLE IX
COMPARISON IN ZSL SETTING

Method	CUB	aPY	SUN	AWA	AWA2
CMT(2013)	37.3	26.9	41.9	58.9	66.3
DEVISE(2013)	53.2	35.4	57.5	72.9	68.6
SJE(2015)	55.3	32.0	57.1	76.7	69.5
SSE(2015)	43.7	31.1	54.5	68.8	67.5
LATEM(2016)	49.4	34.5	56.9	74.8	68.7
ALE(2016)	53.2	30.9	59.1	78.6	80.3
SYNC(2016)	54.1	39.7	59.1	72.2	71.2
LRKT(2016)	46.2	36.8	47.1	81.3	79.2
SAE(2017)	33.4	8.3	42.4	80.6	80.7
GFZSL(2017)	53.0	51.3	**62.9**	80.6	79.3
SCCT(2018)	48.5	–	62.7	74.6	–
ASTE(2018)	49.6	47.4	–	80.8	–
ZSFGC(2019)	49.5	–	–	–	–
Ours+Res.	53.2	48.1	52.7	**82.9**	84.8
Ours+Doub.	**57.3**	**53.7**	55.6	–	**86.7**

Res. denotes the features of ResNet101 are utilized, and Doub. denotes both of the features of ResNet101and InceptionV3 are utilized. We measure top-1 accuracy in %.

method, both the transfer and increment stages are designed to be linear and easy for implementation. Especially, the exemplar generation for the unseen classes requires only the class-level information of the seen classes rather than images, which means our approach does not need to save the training exemplars for the ZSL task at all. To obtain the results shown in Table IX, the training time is only 17.32 and 41.54 s for the aPY and AWA2 data sets (double features), respectively, which is much faster than SE or other network-based methods.

V. CONCLUSION

In this article, we present a transfer-increment mechanism for the GZSL, which has a few requirements for the data and storage resources. Our strategy consists of two stages. First, a dual-knowledge-source-based generative model is developed to synthesize exemplars for the unseen classes and tackle the missing data problem. The generative model is linear and easy for implementation. Second, two training modes, i.e., IWM and IOM, are designed to learn from these virtual exemplars incrementally. Empirically, IOM presents the highest harmonic mean results, while IWM have good recognition ability for the seen classes. With the simplicity and effectiveness, our method is validated on five benchmark data sets and can be further applied in industry and other fields. In addition, potential work can be done to allow the transfer model to learn from the sample-level attributes of the seen data to provide the fine-grained information for the synthesizing process.

REFERENCES

[1] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[4] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 951–958.

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

[6] W. L. Chao, S. Changpinyo, B. Gong, and F. Sha, "An empirical study and analysis of generalized zero-shot learning for object recognition in the wild," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 52–68.

[7] Y. Xian, B. Schiele, and Z. Akata, "Zero-shot learning—The good, the bad and the ugly," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3077–3086.

[8] M. Palatucci, D. Pomerleau, G. E. Hinton, D. Pomerleau, and T. M. Mitchell, "Zero-shot learning with semantic output codes," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2009, pp. 1410–1418.

[9] X. Wang, Y. Ye, and A. Gupta, "Zero-shot recognition via semantic embeddings and knowledge graphs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1211–1219.

[10] S. Bengio, J. Weston, and D. Grangier, "Label embedding trees for large multi-class tasks," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2010, pp. 163–171.

[11] U. Sandouk and K. Chen, "Multi-label zero-shot learning via concept embedding," 2016, *arXiv:1606.00282*. [Online]. Available: http://arxiv.org/abs/1606.00282

[12] L. Zhang *et al.*, "Towards effective deep embedding for zero-shot learning," 2018, *arXiv:1808.10075*. [Online]. Available: http://arxiv.org/abs/1808.10075

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

[14] G. Lin, Y. Chen, and F. Zhao, "Structure propagation for zero-shot learning," 2017, *arXiv:1711.09513*. [Online]. Available: http://arxiv.org/abs/1711.09513

[15] S. Deutsch, S. Kolouri, K. Kim, Y. Owechko, and S. Soatto, "Zero shot learning via multi-scale manifold regularization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5292–5299.

[16] M. Meng and J. Yu, "Zero-shot learning via robust latent representation and manifold regularization," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1824–1836, Apr. 2019.

[17] A.-X. Li, K.-X. Zhang, and L.-W. Wang, "Zero-shot fine-grained classification by deep feature learning with semantics," *Int. J. Autom. Comput.*, vol. 16, no. 5, pp. 563–574, Oct. 2019.

[18] Y. Chen, Y. Xiong, X. Gao, and H. Xiong, "Structurally constrained correlation transfer for zero-shot learning," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2018, pp. 1–4.

[19] S. Rahman, S. Khan, and F. Porikli, "A unified approach for conventional zero-shot, generalized zero-shot, and few-shot learning," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5652–5667, Nov. 2018.

[20] H. Zhang, Y. Long, Y. Guan, and L. Shao, "Triple verification network for generalized zero-shot learning," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 506–517, Jan. 2019.

[21] Y. Wang, H. Zhang, Z. Zhang, and Y. Long, "Asymmetric graph based zero shot learning," *Multimedia Tools Appl.*, vol. 7, pp. 1–22, May 2019.

[22] Y. Yu, Z. Ji, J. Guo, and Z. Zhang, "Zero-shot learning via latent space encoding," *IEEE Trans. Cybern.*, vol. 49, no. 10, pp. 3755–3766, Oct. 2019.

[23] F. Jurie, M. Bucher, and S. Herbin, "Generating visual representations for zero-shot classification," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 2666–2673.

[24] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha, "Synthesized classifiers for zero-shot learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5327–5336.

[25] Y. Guo, G. Ding, J. Han, and Y. Gao, "Synthesizing samples for zero-shot learning," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1774–1780.

[26] D. Wang, Y. Li, Y. Lin, and Y. Zhuang, "Relational knowledge transfer for zero-shot learning," in *Proc. AAAI Conf. Artif. Intell.*, Mar. 2016, pp. 1–7.

[27] B. Zhao, B. Wu, T. Wu, and Y. Wang, "Zero-shot learning posed as a missing data problem," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 2616–2622.

[28] V. K. Verma, G. Arora, A. Mishra, and P. Rai, "Generalized zero-shot learning via synthesized examples," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4281–4289.

[29] V. D. M. Laurens and G. E. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 2605, pp. 2579–2605, 2008.

[30] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1063–6919.

[31] Y. Yu, Z. Ji, J. Guo, and Y. Pang, "Transductive zero-shot learning with adaptive structural embedding," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4116–4127, Sep. 2018.
[32] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-embedding for attribute-based classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 819–826.
[33] A. Frome *et al.*, "Devise: A deep visual-semantic embedding model," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2013, pp. 2121–2129.
[34] B. Romera-Paredes and P. H. S. Torr, "An embarrassingly simple approach to zero-shot learning," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2015, pp. 2152–2161.
[35] R. Socher, M. Ganjoo, C. D. Manning, and A. Y. Ng, "Zero-shot learning through cross-modal transfer," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2013, pp. 935–943.
[36] E. Schonfeld, S. Ebrahimi, S. Sinha, T. Darrell, and Z. Akata, "Generalized zero- and few-shot learning via aligned variational autoencoders," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8247–8255.
[37] Z. Ji, Y. Sun, Y. Yu, Y. Pang, and J. Han, "Attribute-guided network for cross-modal zero-shot hashing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 1, pp. 321–330, Jan. 2020.
[38] Z. Fu, T. A. Xiang, E. Kodirov, and S. Gong, "Zero-shot object recognition by semantic manifold distance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2635–2644.
[39] L. Yang *et al.*, "Zero-shot learning using synthesized unseen visual data with diffusion regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 40, pp. 2498–2512, Oct. 2018.
[40] H. Huang, C. Wang, P. S. Yu, and C.-D. Wang, "Generative dual adversarial network for generalized zero-shot learning," 2018, *arXiv:1811.04857*. [Online]. Available: http://arxiv.org/abs/1811.04857
[41] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, "Feature generating networks for zero-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5542–5551.
[42] Y. Zhu, M. Elhoseiny, B. Liu, X. Peng, and A. Elgammal, "A generative adversarial approach for zero-shot learning from noisy texts," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1004–1013.
[43] Y. Xian, S. Sharma, B. Schiele, and Z. Akata, "F-VAEGAN-d2: A feature generating framework for any-shot learning," 2019, *arXiv:1903.10132*. [Online]. Available: http://arxiv.org/abs/1903.10132
[44] J. Li, M. Jin, K. Lu, Z. Ding, L. Zhu, and Z. Huang, "Leveraging the invariant side of generative zero-shot learning," 2019, *arXiv:1904.04092*. [Online]. Available: http://arxiv.org/abs/1904.04092
[45] N. Xue, Y. Wang, X. Fan, and M. Min, "Incremental zero-shot learning based on attributes for image classification," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 850–854.
[46] P. Kankuekul, A. Kawewong, S. Tangruamsub, and O. Hasegawa, "Online incremental attribute-based zero-shot learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3657–3664.
[47] E. Ferreira, B. Jabaian, and F. Lefevre, "Online adaptative zero-shot learning spoken language understanding using word-embedding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 5321–5325.
[48] M. Rosenblatt, "A central limit theorem and a strong mixing condition," *Proc. Nat. Acad. Sci. USA*, vol. 42, no. 1, pp. 43–47, 1956.
[49] C. Kipnis and S. R. S. Varadhan, "Central limit theorem for additive functionals of reversible Markov processes and applications to simple exclusions," *Commun. Math. Phys.*, vol. 104, no. 1, pp. 1–19, Mar. 1986.
[50] T. Mikosch, A. W. V. D. Vaart, and J. A. Wellner, "Weak convergence of empirical processes," *J. Amer. Stat. Assoc.*, vol. 92, no. 438, p. 794, 1997.
[51] J. A. Wellner, "A Glivenko-Cantelli theorem and strong laws of large numbers for functions of order statistics," *Ann. Statist.*, vol. 5, no. 3, pp. 473–480, 1977.
[52] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *J. Mach. Learn. Res.*, vol. 7, pp. 551–585, Dec. 2006.
[53] L. Wang, H.-B. Ji, and Y. Jin, "Fuzzy passive–aggressive classification: A robust and efficient algorithm for online classification problems," *Inf. Sci.*, vol. 220, pp. 46–63, Jan. 2013.
[54] C.-C. Chang, Y.-J. Lee, and H.-K. Pao, "A passive-aggressive algorithm for semi-supervised learning," in *Proc. Int. Conf. Technol. Appl. Artif. Intell.*, Nov. 2010, pp. 335–341.
[55] P. Welinder *et al.*, "Caltech-UCSD birds-200," Caltech, Pasadena, CA, USA, Tech. Rep. CNS-TR-2010-001, 2010.
[56] G. Patterson and J. Hays, "SUN attribute database: Discovering, annotating, and recognizing scene attributes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2751–2758.
[57] M. Norouzi *et al.*, "Zero-shot learning by convex combination of semantic embeddings," 2013, *arXiv:1312.5650*. [Online]. Available: http://arxiv.org/abs/1312.5650
[58] E. Kodirov, T. Xiang, and S. Gong, "Semantic autoencoder for zero-shot learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4447–4456.
[59] A. Mishra, M. S. K. Reddy, A. Mittal, and H. A Murthy, "A generative model for zero shot learning using conditional variational autoencoders," 2017, *arXiv:1709.00663*. [Online]. Available: http://arxiv.org/abs/1709.00663
[60] Z. Zhang and V. Saligrama, "Zero-shot recognition via structured prediction," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2016, pp. 533–548.
[61] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele, "Evaluation of output embeddings for fine-grained image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2927–2936.
[62] Z. Zhang and V. Saligrama, "Zero-shot learning via semantic similarity embedding," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4166–4174.
[63] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele, "Latent embeddings for zero-shot classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 69–77.
[64] V. K. Verma and P. Rai, "A simple exponential family framework for zero-shot learning," 2017, *arXiv:1707.08040*. [Online]. Available: http://arxiv.org/abs/1707.08040
[65] W. Li, C. Yang, and S. E. Jabari, "Nonlinear traffic prediction as a matrix completion problem with ensemble learning," 2020, *arXiv:2001.02492*. [Online]. Available: http://arxiv.org/abs/2001.02492

Liangjun Feng received the B.Eng. degree from North China Electric Power University, Beijing, China, in 2017. He is currently pursuing the Ph.D. degree with the College of Control Science and Engineering, Zhejiang University, Hangzhou, China.

During the time, he researched at the laboratory of Guotian Yang, studied as an Exchange Student with the University of Wisconsin–Milwaukee, Milwaukee, WI, USA, and got the Beijing Outstanding Graduate Honor. His current research interests include machine learning, artificial intelligence, and pattern recognition.

Chunhui Zhao (Senior Member, IEEE) received the B.Eng., M.Sc., and Ph.D. degrees from Northeastern University, Shenyang, China, in 2003, 2006, and 2009, respectively.

From January 2009 to January 2012, she was a Post-Doctoral Fellow with The Hong Kong University of Science and Technology, Hong Kong, and the University of California at Santa Barbara, Los Angeles, CA, USA. Since January 2012, she has been a Professor with the College of Control Science and Engineering, Zhejiang University, Hangzhou, China. She has authored or coauthored more than 120 articles in the peer-reviewed international journals. Her research interests include statistical machine learning and data mining for industrial application.

Dr. Zhao was a recipient of the National Top 100 Excellent Doctor Thesis Nomination Award, New Century Excellent Talents in University, China, and the National Science Fund for Excellent Young Scholars.