Incremental Zero-Shot Learning

Kun Wei, Cheng Deng, *Senior Member, IEEE*, Xu Yang, *Member, IEEE*, and Dacheng Tao, *Fellow, IEEE*

***Abstract*—The goal of zero-shot learning (ZSL) is to recognize objects from unseen classes correctly without corresponding training samples. The existing ZSL methods are trained on a set of predefined classes and do not have the ability to learn from a stream of training data. However, in many real-world applications, training data are collected incrementally; this is one of the main reasons why ZSL methods cannot be applied to certain real-world situations. Accordingly, in order to handle practical learning tasks of this kind, we introduce a novel ZSL setting, referred to as incremental ZSL (IZSL), the goal of which is to accumulate historical knowledge and alleviate Catastrophic Forgetting to facilitate better recognition when incrementally trained on new classes. We further propose a novel method to realize IZSL, which employs a generative replay strategy to produce virtual samples of previously seen classes. The historical knowledge is then transferred from the former learning step to the current step through joint training on both real new and virtual old data. Subsequently, a knowledge distillation strategy is leveraged to distill the knowledge from the former model to the current model, which regularizes the training process of the current model. In addition, our method can be flexibly equipped with the most generative-ZSL methods to tackle IZSL. Extensive experiments on three challenging benchmarks indicate that the proposed method can effectively tackle the IZSL problem effectively, while the existing ZSL methods fail.**

***Index Terms*—Generative replay, incremental learning, knowledge distillation, transfer knowledge, zero-shot learning (ZSL).**

I. Introduction

In RECENT years, zero-shot learning (ZSL) [1]–[4] has attracted significant attention in computer vision fields [5]–[9]. These methods aim to recognize unseen classes without any labeled training data. In the popular ZSL setting [10], [11], the model is trained on a set of predefined seen classes, which subsequently leverages the learned mapping to transfer the knowledge from seen to unseen classes, the labels of which are disjoint with those of seen classes.

However, the existing ZSL methods [12]–[17] are unable to learn and accumulate the knowledge of new training data sequentially, which leads to their inflexibility in many real-world applications. In contrast, humans have the ability to incrementally learn from a stream of new data and make better predictions on unseen class data when the more classes they see. Inspired by this phenomenon, we propose a more realistic ZSL setting, namely, incremental ZSL (IZSL), the goal of which is to accumulate historical knowledge to improve its ability to recognize unseen classes, as well as to alleviate Catastrophic Forgetting and preserve its ability to recognize seen classes when incrementally learning new classes. As illustrated in Fig. 1, the model is trained on multiple learning steps; each of these steps includes images and semantic embeddings of new classes, which contain abundant information regarding the corresponding classes. The classes of different learning steps are disjoint. During every learning step, the model is evaluated on test images from both seen and unseen classes. The key difference between the traditional ZSL and IZSL settings is that our IZSL involves multiple learning steps on new classes, while the traditional setting only conducts offline training once on a fixed training set.

Due to the rapid growth of online data and constraints on computational resources, incremental learning [18]–[20], which aims to learn incrementally from a stream of training data, has attracted significant attention in recent years. Our IZSL is also a kind of Incremental Learning. However, unlike the mainstream incremental learning setting [21], the training and test sets in our IZSL are disjoint, which is more similar to the human learning process and more suitable for real-world applications. In addition, our setting focuses not only on mitigating Catastrophic Forgetting [22] to preserve the ability to recognize seen classes but also on accumulating historical knowledge to improve the predictions on unseen classes.

Typical ZSL methods [1], [23], [24] try to learn a mapping function between image features and semantic embeddings of given seen class data. Subsequently, given a set of test data, it is projected to semantic embedding space with the learned mapping, and its label is predicted according to the distance to the semantic embeddings of unseen classes. There are also some ZSL methods [2], [10], [25] that train generative models in order to generate corresponding visual features, given the semantic embeddings of the unseen classes, after which the visual features from both seen and unseen classes are utilized to train a better classifier that converts ZSL to Supervised Learning. However, these ZSL methods are unable to deal effectively with the IZSL problem; this is because they cannot accumulate knowledge from old classes that have previously been trained, which limits the application of these methods to real-world situations. For example, as shown in Fig. 2, a pretrained ZSL classifier for animals must recognize the unseen class "panada," which is not contained in

Manuscript received 23 April 2021; accepted 20 August 2021. Date of publication 30 September 2021; date of current version 18 November 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62132016, Grant 62171343, and Grant 62071361; in part by the Key Research and Development Program of Shaanxi under Grant 2021ZDLGY01-03; and in part by the Fundamental Research Funds for the Central Universities under Grant ZDRC2102. This article was recommended by Associate Editor B. Ribeiro. *(Corresponding author: Cheng Deng.)*

Kun Wei, Cheng Deng, and Xu Yang are with the School of Electronic Engineering, Xidian University, Xi'an 710071, China (e-mail: weikunsk@gmail.com; chdeng.xd@gmail.com; xuyang.xd@gmail.com).

Dacheng Tao is with JD Explore Academy, Beijing, China (e-mail: dacheng.tao@gmail.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2021.3110369.

Digital Object Identifier 10.1109/TCYB.2021.3110369

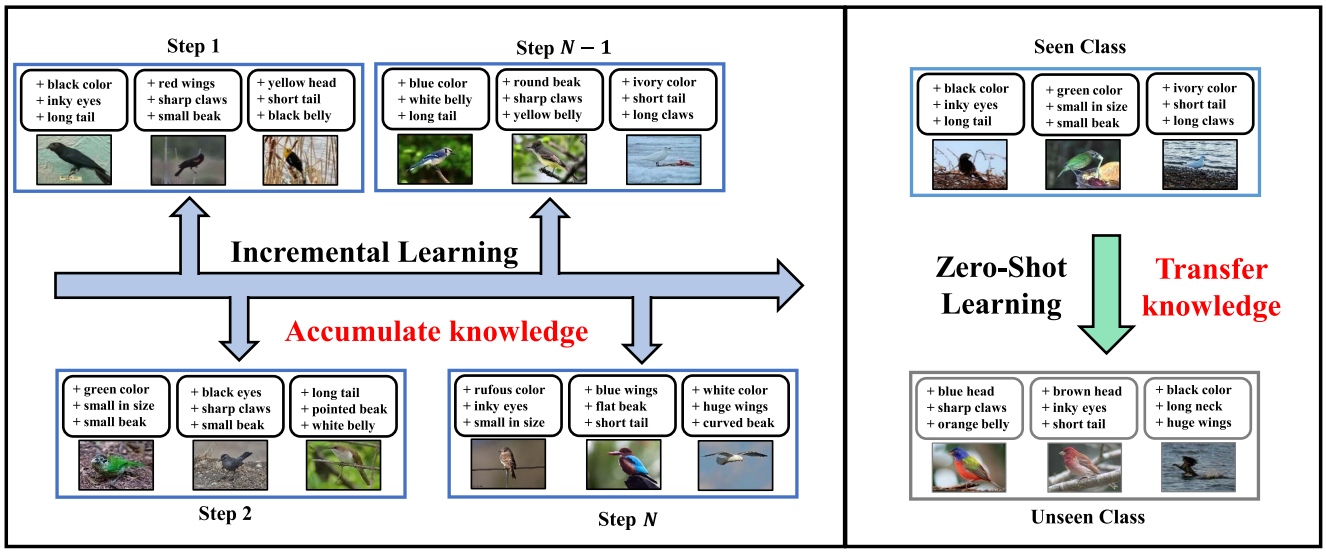

Fig. 1. Overview of IZSL. As the number of seen classes increases, the model learns the new classes sequentially, which accumulates the knowledge from previous seen classes, then transfers the knowledge from seen to unseen classes, which aids the model in effectively classifying these unseen classes.

the training dataset. The training dataset only contains the seen class "bear," which has some attributes related to panada. Moreover, when we collect the data of other related classes, for example, "cat" and "palm civet," existing ZSL methods are unable to capture the knowledge of these classes and cannot recognize the unseen class panada without relearning previous seen classes. For another example, Google employs "Zero-Shot Translation" to achieve the translation between language pairs that have never been seen explicitly, which extends their previous Google neural machine translation (GNMT) to facilitate translation between multiple languages. However, it is still necessary to learn new words and sentences sequentially to improve Zero-Shot translation quality, thus equipping the ZSL methods with the ability to perform incremental learning is essential to the application of these methods in many real-world situations.

Accordingly, in this article, we present a novel approach that employs the generative replay strategy [26], [27] and knowledge distillation strategy [28]. f-CLSWGAN [10], which is made up of traditional ZSL methods combined with generative models, is selected as the basic model. f-CLSWGAN employs Wasserstein-GAN (WGAN) [29] to generate the virtual visual features of unseen classes from the corresponding semantic embedding, after which these features are combined with the visual features of the seen class to train a classifier. Based on f-CLSWGAN, we further introduce the generative replay strategy, which learns to generate the virtual visual features of previously learned classes. The model is then trained jointly on the training data of the current learning step and the virtual data of previously learned classes. In addition, a knowledge distillation strategy is employed to equip the current model with the ability obtained from previous training steps, which ensures that the current model generates similar visual features as the former model generated with the same input.

To evaluate our new method and existing ZSL methods under the proposed IZSL setting, we further design and provide benchmarks tailored to the IZSL problem, based on three popular ZSL datasets. More specifically, we split the seen classes in every dataset into T subsets, each of which has disjoint classes. Each subset serves as the training set in a learning step. Extensive experiments demonstrate that our method is able to effectively accumulate historical knowledge from previously learned classes and alleviate Catastrophic Forgetting, in cases where other state-of-the-art ZSL methods are inoperative. In summary, the contributions of this work are as follows.

1) To the best of our knowledge, we are the first to propose and tackle the IZSL problem. IZSL benchmarks are accordingly designed and provided for evaluation.
2) We devise a novel approach for IZSL that employs a generative replay strategy and knowledge distillation strategy. The generative replay strategy enables knowledge to be transferred from previously learned classes to the current training step, while the knowledge distillation strategy is employed to teach the current model to capture the knowledge of previous learned classes.
3) Extensive experimental results on three benchmarks demonstrate the effectiveness of our proposed approach, which notably outperforms state-of-the-art ZSL methods.

II. Related Work

A. Generative Adversarial Networks

Generative adversarial networks (GANs) [30], which have recently attracted considerable attention in the computer vision field, consist of two components: 1) a generator and 2) a discriminator. The generator learns to map from a latent space to a particular data distribution of interest, while the discriminator distinguishes between samples from the true data distribution and candidates produced by the generator. In order to make full use of the prior information, Mirza and Osindero [31]

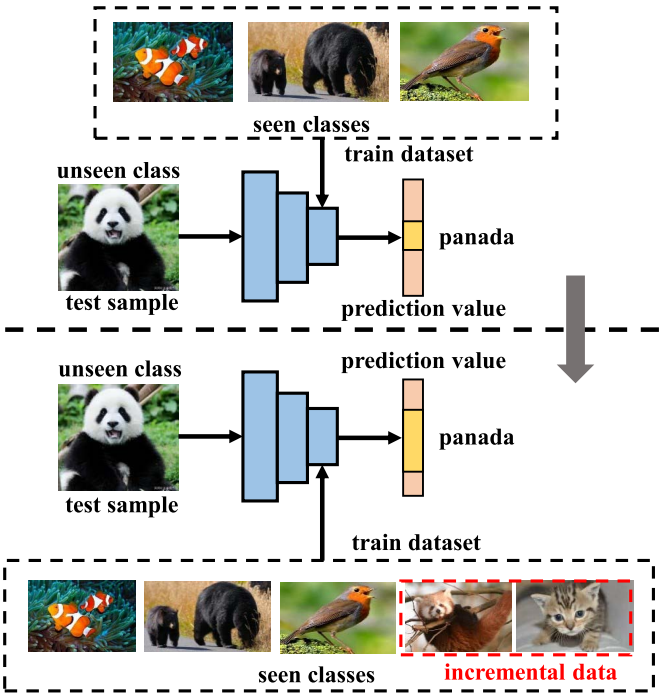

Fig. 2. Real-world example of IZSl. With the classes for training increasing, the prediction of unseen class becomes more accurate.

introduced the conditional GAN (CGAN), which can feed the class labels and sentence descriptions into both the generator and the discriminator. Moreover, in an attempt, to improve the performance of GAN, the seminal technique known as DC-GAN [32] was introduced to stabilize the training of GAN, and has improved performance on many computer vision tasks, for example, image generation [12] and video generation [33]. Subsequently, to solve the unstable training issues associated with GAN, WGAN [29] was proposed to optimize an efficient approximation of the Wasserstein distance. Furthermore, to efficiently solve the task of image-to-image translation, pix2pix [34] was introduced along with its novel generator and discriminator. In addition, to resolve the dilemma arising from the lack of paired data, CycleGANs [35] was introduced based on CGANs and has obtained good performance.

In a departure from the methods described above, our method endows GAN with more new properties, accumulating historical knowledge from the seen classes and transferring this knowledge from the seen to unseen classes, which is the most important bridge to seamlessly connect ZSL and Incremental Learning.

B. Zero-Shot Learning

ZSL is a hot topic in transfer learning, which handles the problem of some test classes not being included in the training set. In the ZSL context, the unseen test samples are captured from the visual space, while their class embeddings are available only in the semantic space. Thus, mainstream ZSL approaches aim to build the connection between the visual and semantic spaces by embedding visual features and semantic embeddings. Typical embedding methods [1], [36] learn a mapping function that projects the visual features into a common embedding space, in which the unseen samples can be recognized. Moreover, AREN [37] pays attention to the region of images, which tries to construct the connection between the regions and attributes. RGEN [38] leverages a region graph embedding network to capture the discriminative information of attributes. VMAN [39] considers the neighborhood relationships between samples in both the semantic and feature spaces. Recent developments have seen the successful introduction of GANs have been successfully introduced to ZSL. The purpose of GAN-based ZSL is to generate visual unseen features from random noise and the corresponding semantic embeddings. For instance, a feature-generating network (f-CLSWGAN) [10] was proposed by employing conditional WGANs. Based on f-CLSWGAN, a new regularization was further employed [25] for GAN training that forces the generated visual features to reconstruct their original semantic embedding. In addition, VAE is employed to synthesize the convolutional neural-network (CNN) features of the unseen classes to tackle the ZSL task by many following methods [40], which obtain impressive performance. Since generative-based methods convert ZSL into a conventional supervised classification problem and achieve appealing performance, we select f-CLSWGAN as our basic model. LZSL [41] is leveraged to alleviate Catastrophic Forgetting when learning different datasets, which cannot be applied in IZSL.

However, all of the above-mentioned methods are trained on a set of predefined classes and, thus, lack the ability to learn additional classes without forgetting knowledge of previous classes. In this article, we thereby propose IZSL to tackle the problem.

C. Incremental Learning

Incremental Learning requires data to arrive sequentially and the transfer of prior knowledge to the current task. A key challenge for Incremental Learning is Catastrophic Forgetting, which refers to cases in which the trained model forgets the previous learned knowledge when adapting to a new task. Many incremental models have been proposed to address this issue, which can be broadly divided into three categories: 1) storing training samples [42]–[44]; 2) regularizing the parameter updates [45]–[48]; and 3) learning a generative model to generate discriminative data [27], [49], [50]. Besides, the replay was first proposed in the work [51], where the images of the previous tasks are produced by generative methods and combined together with the data for the new task by forming a new complete dataset, after which a new model is trained using this new dataset. In addition, some methods [48], [52], [53] leveraged representation learning to bridge the semantic gap between two adjacent tasks. Recently, Incremental Learning has been applied to many computer vision tasks, including Few-Shot Learning [54] and semantic segmentation [55], [56], which bridge the gap between the computer vision field and real-world situations.

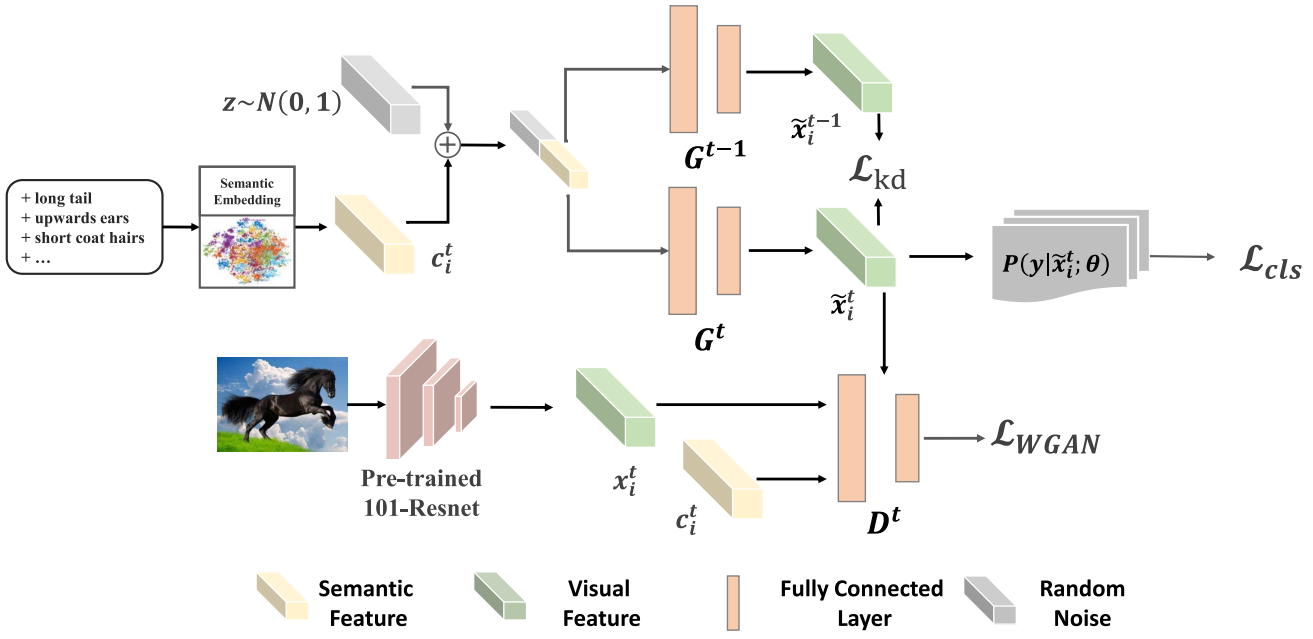

Fig. 3. Framework of our proposed method in the t-th training step. The framework consists of a pretrained feature extractor, a pretrained classifier, two generators, and a discriminator. Given an image, the extractor captures its visual features x_i^t. Meanwhile, the corresponding attribute c_i^t concatenated with random Gaussian noise is mapped as $\tilde{x}_i^{t-1}$ and $\tilde{x}_i^t$ by G^{t-1} and G^t, respectively. Subsequently, the $\tilde{x}_i^t$ is fed into the pretrained classifier and regularized by $\mathcal{L}_{\text{cls}}$. x_i^t and $\tilde{x}_i^t$ are regularized by $\mathcal{L}_{\text{WGAN}}$, which makes x_i^t and $\tilde{x}_i^t$ similar. Finally, G^{t-1} teaches the training process of G^t, supervised by $\mathcal{L}_{kd}$.

Unlike in traditional incremental tasks, the training and test sets are disjoint in our IZSL when others are the same. In this article, we leverage the generative replay strategy to tackle IZSL without the need to train an extra generator, meaning that this approach combines seamlessly with the generative ZSL model.

III. Methodology

The existing ZSL models are only trained once on a set of predefined classes and do not have the ability to incrementally learn new classes without forgetting previously obtained knowledge. To address this problem, we propose IZSL, the framework of which is illustrated as Fig. 3. We leverage a generative replay strategy to transfer the knowledge of seen classes from previous training steps to the current training step, facilitating the sequential accumulation of knowledge. Moreover, we employ a knowledge distillation strategy to supervise the learning process of the current model by encouraging the two networks to produce similar output given the same input, which is effective in alleviating Catastrophic Forgetting.

A. Problem Formulation

Assume that the entire training set is divided into T parts without any overlap. During the t-th training step, we are given a training dataset containing N^t samples, that is, denoted as $D^t = \{(x_i^t, y_i^t, c_i^t)|x_i^t \in X^t, y_i^t \in Y^t, c_i^t \in C^t\}_{i=1}^{N^t}$, where $x_i^t \in X^t$ is the visual feature of t-th image extracted from the CNN, $y_i^t \in Y^t$ denotes the class label in $Y^t = \{y^{1,t}, \ldots, y^{K^t,t}\}$ consisting of K^t seen classes in the t-th training step, and $c_i^t \in C^t$ is the semantic embedding of the class, which is the attribute of class y_i^t. In addition, we have a disjoint class label set of unseen classes $Y_u = \{u_1, \ldots, u_L\}$, whose semantic embedding set $\mathcal{U} = \{(u, c)|u \in y_u, c \in C_u\}$ is available, although the training images and visual features are missing. For all T training steps, Y_u is the same. Given D^t and $\mathcal{U}$, the task of IZSL is to learn a classifier $f_{izsl} : X^t \rightarrow Y_s \cup Y_u$, where $Y_s = (y^{1,1}, y^{2,1}, \ldots, y^{K^t-1,t}, y^{K^t,t})$, which contains all seen classes from the beginning to the current training step. Moreover, we also have all semantic embeddings from the beginning to the current training step, which take the form of point data for each class and can be saved with limited memory.

B. Background: f-CLSWGAN

In this section, we first introduce the f-CLSWGAN model [10], which is the backbone of our approach. The f-CLSWGAN model is composed of a generator G, which produces a virtual visual feature $\tilde{x}$, given its corresponding semantic embedding c and a random Gaussian noise $z \sim N(0, 1)$, a discriminative model D that tries to distinguish the visual feature x from the virtual visual feature $\tilde{x}$ generated from its semantic embedding c, and a pretrained classifier. The original GAN, conditioned on the semantic embedding c, is learned by optimizing the following objective function:

$$\mathcal{L}_{\text{GAN}} = \mathbb{E}\big[\log D(x, c)\big] + \mathbb{E}\big[\log(1 - D(\tilde{x}, c))\big] \quad (1)$$

with $\tilde{x} = G(z, c)$. Based on this, f-CLSWGAN relies on one of the most stable training strategies for GANs, namely, the Wasserstein GAN [29], the objective function of which is denoted as follows:

$$\mathcal{L}_{\text{WGAN}} = \mathbb{E}[D(x,c)] - \mathbb{E}[D(\widetilde{x},c)] - \lambda E\Big[(\|\nabla_{\widehat{x}} D(\widehat{x},c)\|_2 - 1)^2\Big] \tag{2}$$

where $\widetilde{x} = G(z,c)$, $\widehat{x} = \alpha x + (1-\alpha)\widetilde{x}$ with $\alpha \sim U(0,1)$, while λ is the penalty coefficient. The first two terms in (2) approximate the Wasserstein distance, and the third term is the gradient penalty that enforces the gradient of D to regularize the straight line between pairs of real and generated points with a unit norm. To guarantee that the virtual visual features will contain discriminative features of classes and will be suited for training a robust classifier, f-CLSWGAN employs the classification loss over the generated features. The classification loss is defined as follows:

$$\mathcal{L}_{\text{cls}} = -\mathbb{E}_{\widetilde{x}\sim p_{\widetilde{x}}}\big[\log P(y|\widetilde{x};\theta)\big] \tag{3}$$

where $\widetilde{x} = G(z,c)$, y is the class label of $\widetilde{x}$, while $P(y|\widetilde{x};\theta)$ denotes the probability of $\widetilde{x}$ predicted with its true class label y. θ is the parameter of a softmax classifier, which is pretrained on the real visual features of seen classes. The classification loss can enforce the generator to construct the discriminative features of seen classes.

The full objective function then becomes

$$\mathcal{L} = \mathcal{L}_{\text{WGAN}} + \beta \mathcal{L}_{\text{cls}} \tag{4}$$

where β is a hyperparameter used to balance the classification loss and the WGAN loss, which is empirically set to 0.01.

In the testing stage, given the semantic embedding c from an unseen class $u \in Y_u$, f-CLSWGAN combines it with Gaussian noise z and generates corresponding virtual visual feature $\widetilde{x}$; this process is then repeated n_{cls} times. We subsequently combine the visual features of the seen classes and the virtual visual features of the unseen classes to construct a new joint dataset, D_{cls}, which is a dataset for supervised classification. Finally, we train another classifier using this new dataset D_{cls}, which is a standard softmax classifier. The standard softmax classifier minimizes the negative log-likelihood loss as follows:

$$\min_{\theta} -\frac{1}{|\Gamma|}\sum_{(x,y)\in\Gamma} \log P(y|x;\theta) \tag{5}$$

where θ is the weight matrix of a fully connected layer that projects the visual feature x to N unnormalized probabilities, with N being the number of candidate classes. Moreover, $\Gamma = \mathcal{S} \cup \widetilde{\mathcal{U}}$ for our IZSL, where $\mathcal{S}$ is denoted as the training data of the seen classes and $\widetilde{\mathcal{U}}$ is defined as the generated data of the unseen classes. The prediction objective function is as follows:

$$f(x) = \arg\max_{y} P(y|x;\theta) \tag{6}$$

where $y \in Y_s \cup Y_u$ in IZSL. The predictions of the test dataset are used to evaluate the performance of the method.

C. Generative Replay

In the t-th training step, the network is trained on D^t without relearning the samples of previous seen classes; this does not accumulate the knowledge from previous seen classes or enable the samples of the unseen classes to be accurately recognized. The effective way to tackle this theory problem is to transfer knowledge from previous training steps to the current training step.

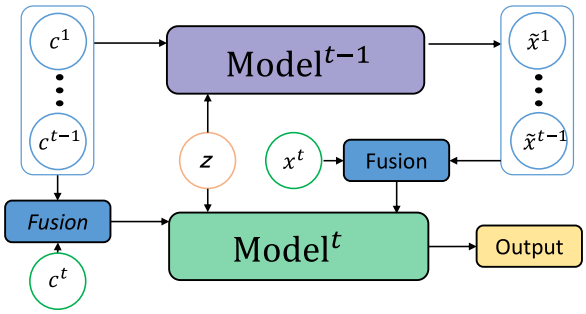

Fig. 4. Illustration of generative replay.

Inspired by previous incremental learning methods [27], [57], we seamlessly integrate the generative replay strategy with f-CLSWGAN seamlessly. Unlike other incremental classification methods, our method does not require an extra generator and transfers more discriminative knowledge of the classes, which is learned in previous training steps. The generative replay strategy is leveraged between two training steps to facilitate knowledge transfer. As shown in Fig. 4, given c_i^m $(0 < m < t)$ and random Gaussian noise z, we employ G^{t-1} to generate the virtual visual feature $\widetilde{x}_i^m$ of the previous seen class, which contains the knowledge of class y_i^m. This generation process is then repeated n_{re} times for every previous seen class, while the features generated via this process are used to construct a generated dataset. Then, by fusing the dataset D^t and the generated dataset, which consists of virtual visual features, semantic embeddings, and class labels, we can obtain the new dataset D^t_{joint}, which contains all the visual features of seen classes from the beginning to the t-th training step. We next use the new dataset D^t_{joint} to complete the training process in the t-th training step. With the aid of the generative replay strategy, in the t-th training step, our method can convert the IZSL problem into a traditional ZSL problem and attain better prior knowledge for use in recognizing unseen classes.

D. Knowledge Distillation

At the t-th steps, we obtain the visual features of previous seen classes, which contain knowledge of previous steps after generative replay is complete. With the goal of accumulating the knowledge of seen classes further, the knowledge distillation strategy is employed to supervise the learning process of the current step. The knowledge distillation strategy is adopted to distill information from a previously trained network to the current network by encouraging these two networks to produce similar output values or patterns when given the same input. As shown in Fig. 3, c_i^m, concatenated with random Gaussian noise z, is fed into G^{t-1} and G^t, respectively, then mapped as $\widetilde{x}_i^{t-1}$ and $\widetilde{x}_i^t$, which are virtual visual features generated by the two generators. To force G^t to obtain the same ability to generate visual features of previous seen classes, we introduce the knowledge distillation loss to align $\widetilde{x}_i^{t-1}$ and $\widetilde{x}_i^t$ in visual

TABLE I
DATASETS USED IN OUR EXPERIMENTS, AND THEIR STATISTICS

Dataset	Semantics Dim	Image	Seen Classes	Unseen Classes
CUB	312	11788	150	50
FLO	1024	8189	82	20
SUN	102	14340	645	72

space. The knowledge distillation loss is denoted as follows:

$$\mathcal{L}_{\mathrm{KD}} = \left\| \widetilde{x}_i^{t-1} - \widetilde{x}_i^{t} \right\|_1. \tag{7}$$

Moreover, when $t > 1$, the objective function is denoted as follows:

$$\mathcal{L} = \mathcal{L}_{\mathrm{WGAN}} + \beta \mathcal{L}_{\mathrm{cls}} + \lambda \mathcal{L}_{\mathrm{KD}} \tag{8}$$

where λ is the hyperparameter used to weight the knowledge distillation loss and is set to 20.

E. Training and Inference

In the training stage, the model is trained sequentially on different split datasets sequentially. The generative replay strategy is employed between two split datasets to generate the discriminative visual features of previous seen classes. The generated visual features are then fused with the current dataset to construct a new dataset for the training process of the current step. Except for the first step, the knowledge distillation process is leveraged to supervise the training process.

In the testing stage, G^T is employed to generate the virtual visual features of all previous seen classes and unseen classes. The generated and current visual features are then fused to train a softmax classifier. All test samples are fed into the trained classifier to obtain the predictions.

IV. EXPERIMENTS

In this section, we first present the benchmarks and implementation details. A comparison with several competitors is then provided. Finally, we conduct ablation studies to prove the effectiveness of the proposed method.

A. Dataset and Evaluation Metrics

We evaluate our method on three popular datasets: 1) the Caltech-UCSD-Birds 200-2011 dataset (CUB) [58]; 2) the Oxford Flowers dataset (FLO) [59]; and 3) the SUN Attribute dataset (SUN) [60]. In terms of the number of classes, these datasets are ranked in the top three out of the six most popular ZSL datasets. Statistics of the datasets are presented in Table I. For all datasets, we extract 2048-D visual features from the entire images using the pretrained 101-layered ResNet [61]; the test classes in our datasets are disjoint with those classes in the ImageNet training set. In terms of semantic embedding, we use per-class attributes for CUB and SUN, while for FLO, we extract 1024-D character-based CNN-RNN [62] features from fine-grained visual descriptions (ten sentences per image).

The seen classes of the three datasets are divided into five parts on average by label order, whose classes are not overlapping. The test seen classes for the t-th training step contain all seen classes from the beginning to the current training step. The test seen classes change with the training step.

When incrementally learning the new seen classes, the goal of IZSL is to accumulate the historical knowledge to better recognize the unseen classes better, as well as to alleviate Catastrophic Forgetting to preserve the ability to recognize seen classes. Therefore, following the Generalized ZSL setting [10], [24], we employ the same evaluation metrics for IZSL.

1) *u:* Average per-class classification accuracy on test images from the unseen classes with the prediction label set, which is used to measure the capacity to recognize unseen classes.
2) *s:* Average per-class classification accuracy on test images from the seen classes with the prediction label set, which is used to measure the capacity to recognize incremental seen classes.
3) *H:* The harmonic mean of u and s, which is formulated as follows:

$$H = \frac{2 \times u \times s}{u + s}. \tag{9}$$

H balances the performance between the u and s metrics, which are the most important metrics for our task. All results of the three metrics are measured after the final training step.

To evaluate our method's ability to accumulate knowledge and alleviate Catastrophic Forgetting, respectively, we select the ZSL metric and the mean accuracy for every seen class rather than u and s in GZSL, which are the balanced values used to achieve the best H results. In ZSL, the test data are only from unseen classes. The ZSL metric means average per-class classification accuracy on test images from the unseen classes with the unseen label set. Moreover, the mean accuracy for every seen class is the traditional metric used in incremental learning methods to evaluate the performance of alleviating Catastrophic Forgetting.

Average forgetting is defined to estimate the forgetting of previous steps. The forgetting for the jth step is $f_j^k = \max_{l \in 1,\ldots,k-1}(a_{l,j} - a_{k,j})\ \forall j < k$. The average forgetting at the kth step is written as $F_k = (1/[k-1]) \sum_{j=1}^{k-1} f_j^k$.

B. Implementation Details

Our method consists of one generator and one discriminator, both of which are MLP with LeakyReLU activation. The hidden layer of the generator consists of 4096 hidden units, while the discriminator has one hidden layer with 4096 units. Our method is implemented with PyTorch[1] and optimized by the ADAM optimizer. The learning rate and batch size are set to 0.0001 and 64, respectively, and the epoch of each training step is set to 100. We perform one update for generator parameters after five discriminator updates. The numbers of generated features for classification n_{cls} and for generative replay n_{re} will

[1] https://pytorch.org/

TABLE II
CLASSIFICATION ACCURACY (%) OF IZSL WITH THE THREE EVALUATION METRICS ON THE THREE DATASETS

Method	CUB			FLO			SUN		
	u	s	H	u	s	H	u	s	H
base	25.19	19.10	21.73	10.42	14.53	12.14	28.19	13.76	18.49
SFT	34.34	34.06	34.19	30.11	33.34	31.64	31.18	20.24	24.40
$L1$	35.75	35.47	35.61	37.09	32.67	34.74	37.86	21.82	27.68
$L2$	36.06	36.27	36.16	32.18	36.46	34.19	36.60	22.75	28.06
EWC	36.58	34.79	35.67	33.80	36.31	35.01	**40.97**	21.55	28.24
MAS	36.71	36.54	36.63	40.57	51.41	45.35	35.21	19.34	24.97
SDC	21.64	**48.19**	29.87	29.44	**72.12**	41.81	14.51	**30.04**	19.57
Ours	**38.12**	45.63	**41.54**	**55.00**	61.10	**57.89**	38.96	24.77	**30.28**
JL	43.70	57.70	49.70	59.00	73.80	65.60	42.60	36.60	39.40

be discussed in the ablation study, the settings of which differ between datasets.

The number of subtraining datasets is flexible and is set to 5 as the IZSL standard in our experiments. In addition, we divide the dataset equally by the original label order into five parts without artificial selection; we hope that such a division strategy will become the standard for IZSL. If we divide the seen classes randomly, the results of our method may be fluctuating, which would not conducive to a fair comparison. Especially, when the number becomes larger, the phenomenon of Catastrophic Forgetting will be more obvious and the performances will decrease.

C. Comparison to Existing ZSL Methods

1) Baseline Models: Since there is no previous work for IZSL, we combine several traditional incremental learning methods with the f-CLSWGAN as the competitor methods. The traditional incremental learning methods are as follows.

1) *Sequential Fine-Tuning (SFT):* The SFT means the model is fine-tuned in a sequential manner with parameters initialized from the model fine-tuned on the previous task.
2) *L1 Regularization (L1):* At each step t, G^t is initialized as G^{t-1} and continuously trained with $L1$- regularization between G^t and G^{t-1}.
3) *L2 Regularization (L2):* At each step t, G^t is initialized as G^{t-1} and continuously trained with $L2$- regularization between G^t and G^{t-1}.
4) *EWC [45]:* This method was proposed to keep the network parameters close to the optimal parameters for the previous step while training the current step.
5) *MAS [63]:* This method was proposed to accumulate an importance measure for each network parameter based on how sensitive the predicted output function is to a change in this parameter.
6) *SDC [52]:* This method aims to approximate the semantic drift of prototypes after training of new step. The method is complementary to several existing incremental learning methods to improve the performance further.

2) Results and Analysis: Table II summarizes the results of all comparison methods and our method under three evaluation metrics on three benchmark datasets. For all datasets, our method significantly improves u, s, and H relative to the baselines. In addition, joint learning (JL) refers to the results when the model is directly trained on the entire training set, which constitutes the upper bound of the performance, while "Base" indicates the results when the model is trained directly without any incremental method. Apart from "JL" and our method, other methods need to generate the virtual visual features of previous seen features with the generator G^T, which are then fused with D^t to train the Softmax classifier. The promising performance of IZSL is expected to achieve satisfactory results in terms of the H metric, which is the most important metric to balance the performance of recognizing seen classes and unseen classes. Compared to the results of JL, the results of other methods decrease in u, s, and H, which indicates the existence of Catastrophic Forgetting in the IZSL setting and the necessity of further studying IZSL. On CUB, our model achieves 38.12% in u, 45.63% in s, and 41.54% in H, with improvements of 1.14% in u and 4.91% in H relative to the competitors. On FLO, our model achieves 55.00% in u, 61.10% in s, and 57.89% in H, with improvements of 14.43% in u and 12.54% in H relative to the competitors. On SUN, our model achieves 38.96% in u, 24.77% in s, and 30.28% in H, with improvements of 2.04% in H relative to the competitors. SDC obtains the best s performance and unsatisfied u performance on three datasets, which proves that SDC cannot balance the recognition on seen and unseen classes. The classification performance improvement is contributed by the effectiveness of our model in accumulating knowledge from previous seen classes and alleviating Catastrophic Forgetting. Moreover, the performance improvement relative to the comparison methods is more significant on the FLO dataset. This phenomenon proves that our method can alleviate the historical knowledge of previous seen classes when semantic embeddings are more fine-grained. Overall, our model achieves a great balance between u and s metrics and outperforms the baseline methods for IZSL by a large margin.

Table III summarizes the results of all comparison methods and our method under ZSL settings on the benchmark

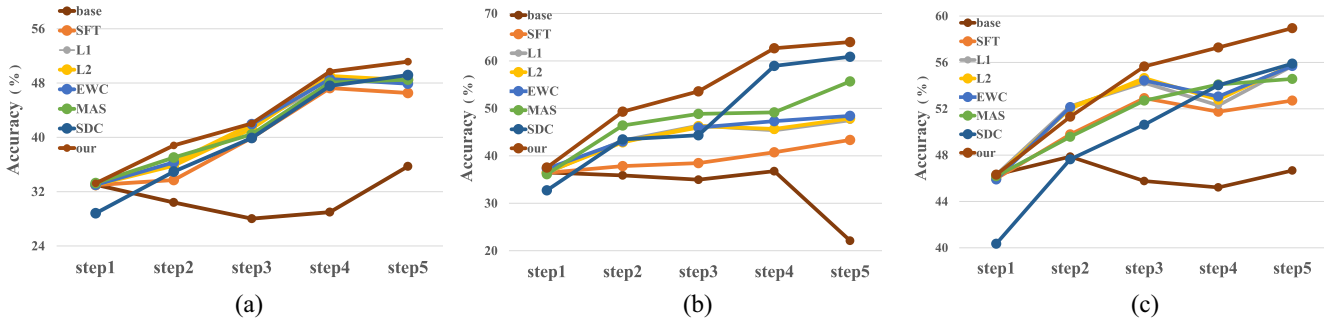

Fig. 5. ZSL results of competitors and our method in five steps on three datasets. (a) CUB. (b) FLO. (c) SUN.

TABLE III
ZSL RESULTS (%) ON IZSL SETTING

Method	CUB	FLO	SUN
base	35.73	22.11	46.67
SFT	46.54	43.31	52.71
$L1$	49.01	47.49	55.69
$L2$	48.42	47.87	55.81
EWC	47.91	48.47	55.69
MAS	48.50	55.69	54.58
SDC	49.17	60.87	55.90
Ours	**51.16**	**63.98**	**58.96**
JL	57.30	67.20	60.80

TABLE IV
MEAN ACCURACIES AND AVERAGE FORGETTING OF SEEN CLASSES (%) ON IZSL SETTING

Method	CUB	FLO	SUN
base	28.54 (65.77)	21.27 (93.33)	18.57 (53.07)
SFT	45.39 (58.22)	46.24 (65.43)	25.27 (49.73)
$L1$	48.38 (55.87)	48.44 (60.15)	28.60 (47.48)
$L2$	48.28 (54.94)	47.63 (62.11)	29.22 (48.06)
EWC	48.82 (55.59)	49.07 (57.33)	28.53 (43.87)
MAS	51.64 (45.17)	64.79 (42.66)	25.93 (45.79)
SDC	51.42 (44.23)	74.51 (39.76)	30.62 (42.91)
Ours	**59.36 (40.39)**	**79.49 (26.68)**	**32.25 (41.33)**
JL	73.06	91.45	48.41

datasets. The ZSL setting is employed to evaluate the ability to accumulate historical knowledge. All methods need to generate virtual visual features of unseen classes, which are then used to train a Softmax classifier. The more knowledge the model accumulates, the more knowledge it transfers from seen to unseen classes, which leads to better predictions under the ZSL setting. On CUB, our model achieves 51.16%, representing an improvement of 1.99% over the competitors. On FLO, our model achieves 63.98%, and 3.11% better than the competitors. On SUN, our model achieves 58.96%, obtaining an improvement of 3.06% relative to the competitors. Our method obtains the best performance on all three datasets, meaning that it demonstrates the best ability to accumulate historical knowledge. In addition, the results of all comparison methods and our method in the five steps are shown in Fig. 5 for ZSL settings on the three benchmark datasets. The performance of the base fluctuates across the five steps, while that of the other methods increases, which means these methods obtain the ability to accumulate knowledge. However, the performances of other methods would be decreased over the training process, while this phenomenon does not appear in our method, which proves the superiority of our proposed approach in accumulating historical knowledge.

Table IV summarizes the accuracies and average forgetting among seen classes of all comparison methods and our method on three benchmark datasets. The accuracies of seen classes are employed to evaluate the ability to alleviate Catastrophic Forgetting. Except for JL, other methods need to generate the virtual visual features of previously seen features with the generator G^t, after which these features are fused with D^t to train the Softmax classifier. The more knowledge the model forgets, the worse performance it yields. On CUB, our model achieves 59.36%, an improvement of 7.72% over the competitors. On FLO, our model achieves 79.49%, an improvement of 4.98% over the competitors. On SUN, our model achieves 32.25%, an improvement of 1.63% over the competitors. The average forgetting results are also shown in Table IV, and our method also obtains the best performance on all three datasets, demonstrating that our method effectively alleviates Catastrophic Forgetting. In addition, the accuracies among seen classes of all comparison methods and our method in five steps are shown in Fig. 6 on the three benchmark datasets. The performances of all methods decrease as the training process progresses, which means the phenomenon of Catastrophic Forgetting still exists. However, our performance reduction trend is the slowest among all methods, which indicates the superiority of our method in alleviating Catastrophic Forgetting.

D. Ablation Study

We conduct ablation experiments to prove the effectiveness of our method.

The results of our basic model with the addition of different modules are presented in Table V. The base model is our model without either the generative replay strategy and knowledge distillation strategy. Based on the base model, we can add the generative replay strategy and knowledge distillation strategy, which are represented as "RP" and "KD," respectively. As shown in Table V, both of these strategies can improve performance in terms of the u, s, and H metrics on the three datasets. The improvement resulting from

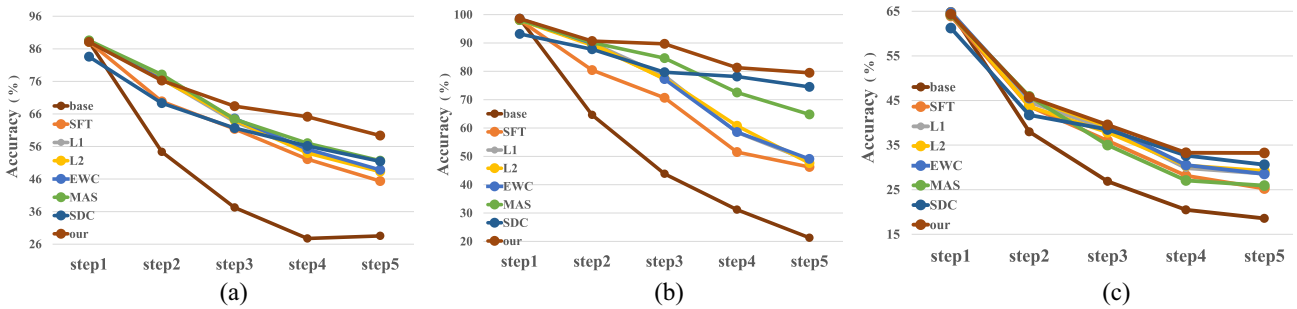

Fig. 6. Mean accuracies of competitors and our method for seen classes in five steps on three datasets. (a) CUB. (b) FLO. (c) SUN.

TABLE V
ABLATION STUDY: CLASSIFICATION ACCURACY (%) WITH DIFFERENT MODULES, "RP" AND "KD," RESPECTIVELY, INDICATE GENERATIVE REPLAY AND KNOWLEDGE DISTILLATION

Method	CUB			FLO			SUN		
	u	s	H	u	s	H	u	s	H
base	25.19	19.10	21.73	10.42	14.53	12.14	28.19	13.76	18.49
+RP	**40.34**	35.95	38.02	48.37	58.38	52.91	38.75	22.95	28.82
+KD	38.16	38.03	38.09	36.24	43.70	39.62	**40.90**	23.60	29.93
+RP+KD	38.12	**45.63**	**41.54**	**55.00**	**61.10**	**57.89**	38.96	**24.77**	**30.28**

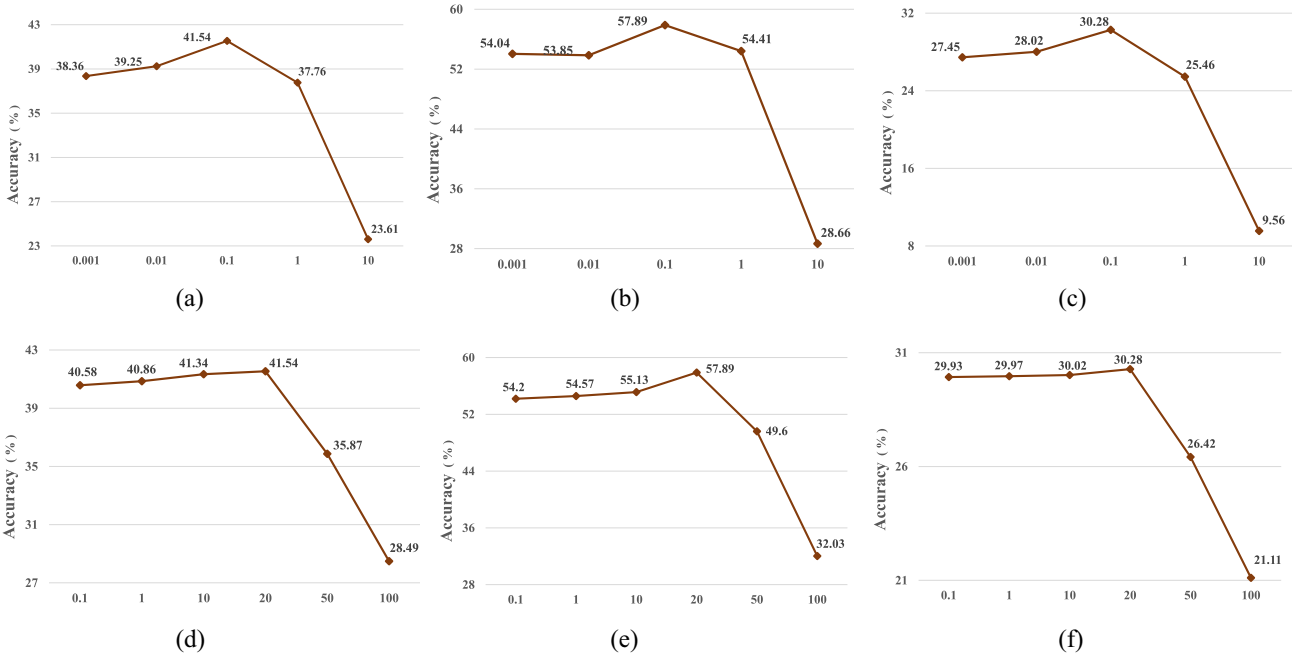

Fig. 7. (a)–(c) Parameter analysis of β on CUB, FLO, and SUN datasets. (d)–(f) Parameter analysis of λ on CUB, FLO, and SUN datasets.

adding RP indicates that the generative replay strategy can transfer the knowledge of the seen classes from previous training steps to the current training step. Moreover, the benefit accorded by generative replay is that the last training step can be viewed as JL. In addition, after adding KD to the base model, better results are obtained for the u, s, and H metrics, meaning that the knowledge distillation strategy effectively preserves the discriminative features of classes when transferring the knowledge from seen to unseen classes. On the FLO dataset, the generative replay strategy brings about more improvement when compared to the knowledge distillation strategy, indicating that the generative replay strategy can accumulate more historical knowledge when semantic embeddings are more fine-grained. When all modules are combined, our method achieves the best performance.

We select CACD-VAE [64] and TF-VAEGAN [65] as the basic generative-ZSL models to apply with our method, whose results on the CUB dataset under the IZSL setting are shown in Table VI. It is obvious that these methods also suffer from Catastrophic Forgetting and obtain unsatisfied recognition performance. When CACD-VAE and TF-VAEGAN are combined with our method, these methods significantly alleviate

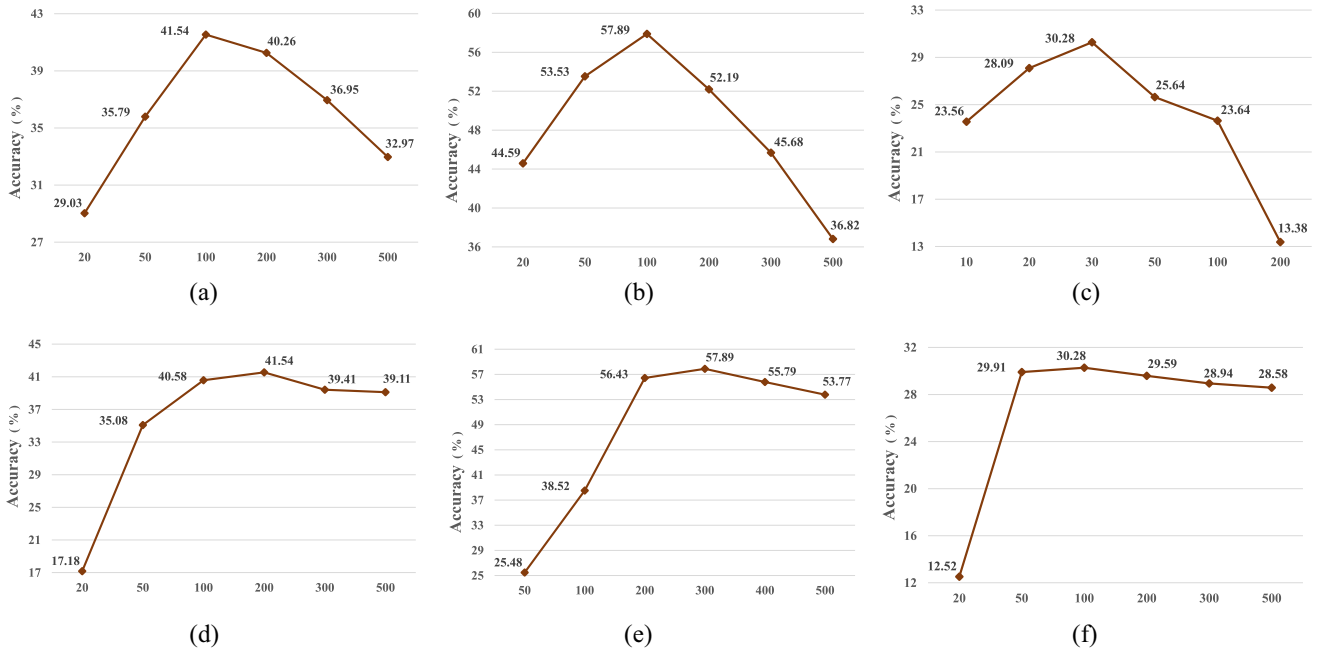

Fig. 8. (a)–(c) Results for H evaluation metric on CUB, FLO, and SUN datasets when increasing the numbers of generative replays n_{re}. (d)–(f) Results for H evaluation metrics on CUB, FLO and SUN datasets when increasing the numbers of classifications n_{cls}.

TABLE VI
ABLATION STUDY: THE RESULTS WHEN OTHER ZSL METHODS ARE APPLIED WITH OUR METHOD

Method	CUB		
	u	s	H
CACD-VAE+SFT	35.19	43.83	39.05
CACD-VAE+IZSL	**38.73**	**50.08**	**43.68**
TF-VAEGAN+SFT	34.22	42.28	37.82
TF-VAEGAN+IZSL	**39.12**	**49.54**	**43.72**

Catastrophic Forgetting and achieve better performance on all three evaluation metrics, which proves the flexibility of our method.

The analyses of hyperparameters are presented in Fig. 7. It is obvious that the proposed method obtains the best performance on three datasets when β and λ are set to 0.1 and 20, respectively. When β is set to 0.1, the generated features can obtain more discriminative and robust information. When λ is set to 20, the proposed method balances the knowledge of previous steps and the current step.

We further perform an experiment to discuss the numbers of generative replays n_{re} and generated samples for classification n_{cls} per class, the results of which are shown in Fig. 8. We select the H metric to evaluate the performance of our method. The number of classes is different for different datasets: specifically, 59 for CUB, 20 for SUN, and 80 for FLO. It is thus better to set a smaller number of n_{re} and n_{cls} for SUN. Therefore, the best performance is achieved when n_{re} and n_{cls} are set to 100 and 300 for CUB, to 100 and 300 for FLO, and 30 and 100 for SUN. Obviously, we find that H increases as n_{re} increases before achieving the peak performance of H. After the peak, H decreases with the increase of n_{re}. This phenomenon indicates that balanced training datasets for all seen classes are better for transferring the knowledge from previous steps to the current step, which leads to an impressive performance in terms of the H metric. In addition, we note that H increases with the increase of n_{cls} for classification, which is essential for knowledge transfer from seen classes to unseen classes. Finally, the performance of the IZSL method is sensitive to these hyperparameters, which means that a balanced dataset is important to facilitate this knowledge transfer.

V. CONCLUSION

To the best of our knowledge, this article represented the first attempt to introduce and tackle IZSL, which better bridges the gap between real-world requirements and computer vision building blocks. A generative replay strategy was employed to accumulate historical knowledge of previously seen classes, which converts IZSL into traditional ZSL. In addition, a knowledge distillation strategy was leveraged to distill the information from the former model to the current model and regularize the current training process, an approach that alleviates Catastrophic Forgetting and facilitates satisfactory recognition performance of seen classes and unseen classes. In addition, our method can be flexibly applied to most generative-based ZSL methods to tackle the IZSL problem. Experiments showed that our method outperforms previous methods by a large margin on three benchmark datasets. An ablation study was also performed to verify that the proposed two strategies are both important to the achievement of good performance.

From the ablation study, we can further note that a balanced dataset can transfer more historical knowledge of seen classes between different steps. An IZSL method with adaptive

generative replay numbers should be proposed in future work to obtain better performance. Moreover, shifting the embeddings between different steps is also a solution to tackling the problem of IZSL and, thus, also represents a promising future research direction.

REFERENCES

[1] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, "Zero-shot learning through cross-modal transfer," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 935–943.

[2] Y. Long, L. Liu, L. Shao, F. Shen, G. Ding, and J. Han, "From zero-shot learning to conventional supervised classification: Unseen visual data synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1627–1636.

[3] E. Kodirov, T. Xiang, Z. Fu, and S. Gong, "Unsupervised domain adaptation for zero-shot learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2452–2460.

[4] X. Xu, H. Lu, J. Song, Y. Yang, H. T. Shen, and X. Li, "Ternary adversarial networks with self-supervision for zero-shot cross-modal retrieval," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2400–2413, Jun. 2020.

[5] D. Chang *et al.*, "The devil is in the channels: Mutual-channel loss for fine-grained image classification," *IEEE Trans. Image Process.*, vol. 20, pp. 4683–4695, 2020.

[6] X. Yang, C. Deng, K. Wei, J. Yan, and W. Liu, "Adversarial learning for robust deep clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 9098–9108.

[7] X. Li *et al.*, "OSLNet: Deep small-sample classification with an orthogonal softmax layer," *IEEE Trans. Image Process.*, vol. 29, pp. 6482–6495, 2020.

[8] X. Yang, C. Deng, Z. Dang, K. Wei, and J. Yan, "SelfSAGCN: Self-supervised semantic alignment for graph convolution network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 16775–16784.

[9] Z. Dang, C. Deng, X. Yang, K. Wei, and H. Huang, "Nearest neighbor matching for deep clustering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 13693–13702.

[10] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, "Feature generating networks for zero-shot learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5542–5551.

[11] 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.

[12] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proc. CVPR*, 2018, pp. 8789–8797.

[13] Y. Yu *et al.*, "Transductive zero-shot learning with a self-training dictionary approach," *IEEE Trans. Cybern.*, vol. 48, no. 10, pp. 2908–2919, Oct. 2018.

[14] 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.

[15] X. Xu, I. W. Tsang, and C. Liu, "Complementary attributes: A new clue to zero-shot learning," *IEEE Trans. Cybern.*, vol. 40, no. 10, pp. 3755–3766, Oct. 2019.

[16] K. Wei, M. Yang, H. Wang, C. Deng, and X. Liu, "Adversarial fine-grained composition learning for unseen attribute-object recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 3741–3749.

[17] X. Li, Z. Xu, K. Wei, and C. Deng, "Generalized zero-shot learning via disentangled representation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 1966–1974.

[18] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 409–415.

[19] M. Hu, Y. Wang, Z. Zhang, D. Zhang, and J. J. Little, "Incremental learning for video-based gait recognition with LBP flow," *IEEE Trans. Cybern.*, vol. 43, no. 1, pp. 77–89, Feb. 2013.

[20] S. Yin, X. Xie, J. Lam, K. C. Cheung, and H. Gao, "An improved incremental learning approach for KPI prognosis of dynamic fuel cell system," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 3135–3144, Dec. 2015.

[21] R. Kemker and C. Kanan, "FearNet: Brain-inspired model for incremental learning," 2017. [Online]. Available: arXiv:1711.10563.

[22] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of Learning and Motivation*. Amsterdam, The Netherlands: Elsevier, 1989, pp. 109–165.

[23] B. Romera-Paredes and P. Torr, "An embarrassingly simple approach to zero-shot learning," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2152–2161.

[24] L. Chen, H. Zhang, J. Xiao, W. Liu, and S.-F. Chang, "Zero-shot visual recognition using semantics-preserving adversarial embedding networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1043–1052.

[25] R. Felix, V. B. Kumar, I. Reid, and G. Carneiro, "Multi-modal cycle-consistent generalized zero-shot learning," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 21–37.

[26] Y. Xiang, Y. Fu, P. Ji, and H. Huang, "Incremental learning using conditional adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 6619–6628.

[27] C. Wu *et al.*, "Memory replay GANs: Learning to generate new categories without forgetting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 5962–5972.

[28] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *Stat*, vol. 1050, p. 9, 2015.

[29] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017. [Online]. Available: arXiv:1701.07875.

[30] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[31] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014. [Online]. Available: arXiv:1411.1784.

[32] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015. [Online]. Available: arXiv:1511.06434.

[33] Y. Li, M. R. Min, D. Shen, D. Carlson, and L. Carin, "Video generation from text," in *Proc. AAAI*, 2018, pp. 1–8.

[34] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1125–1134.

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

[36] L. Zhang, T. Xiang, and S. Gong, "Learning a deep embedding model for zero-shot learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2021–2030.

[37] G.-S. Xie *et al.*, "Attentive region embedding network for zero-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9384–9393.

[38] G.-S. Xie *et al.*, "Region graph embedding network for zero-shot learning," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 562–580.

[39] G.-S. Xie, X.-Y. Zhang, Y. Yao, Z. Zhang, F. Zhao, and L. Shao, "VMAN: A virtual mainstay alignment network for transductive zero-shot learning," *IEEE Trans. Image Process.*, vol. 30, pp. 4316–4329, 2021.

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

[41] K. Wei, C. Deng, and X. Yang, "Lifelong zero-shot learning," in *Proc. IJCAI*, 2020, pp. 551–557.

[42] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCARL: Incremental classifier and representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2001–2010.

[43] Y. Liu, Y. Su, A.-A. Liu, B. Schiele, and Q. Sun, "Mnemonics training: Multi-class incremental learning without forgetting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12245–12254.

[44] J. Rajasegaran, S. Khan, M. Hayat, F. S. Khan, and M. Shah, "iTAML: An incremental task-agnostic meta-learning approach," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13588–13597.

[45] J. Kirkpatrick *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, 2017.

[46] X. Liu, M. Masana, L. Herranz, J. Van de Weijer, A. M. Lopez, and A. D. Bagdanov, "Rotate your networks: Better weight consolidation and less catastrophic forgetting," in *Proc. IEEE 24th Int. Conf. Pattern Recognit. (ICPR)*, 2018, pp. 2262–2268.

[47] J. Zhang *et al.*, "Class-incremental learning via deep model consolidation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2020, pp. 1131–1140.

[48] K. Wei, C. Deng, X. Yang, and M. Li, "Incremental embedding learning via zero-shot translation," 2020. [Online]. Available: arXiv:2012.15497.
[49] A. Robins, "Catastrophic forgetting, rehearsal and pseudorehearsal," *Connection Sci.*, vol. 7, no. 2, pp. 123–146, 1995.
[50] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2990–2999.
[51] A. Seff, A. Beatson, D. Suo, and H. Liu, "Continual learning in generative adversarial nets," 2017.
[52] L. Yu *et al.*, "Semantic drift compensation for class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6982–6991.
[53] B. Zhao, X. Xiao, G. Gan, B. Zhang, and S.-T. Xia, "Maintaining discrimination and fairness in class incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13208–13217.
[54] X. Tao, X. Hong, X. Chang, S. Dong, X. Wei, and Y. Gong, "Few-shot class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12183–12192.
[55] U. Michieli and P. Zanuttigh, "Incremental learning techniques for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2019, pp. 3205–3212.
[56] Y. Gu, C. Deng, and K. Wei, "Class-incremental instance segmentation via multi-teacher networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 1478–1486.
[57] M. Zhai, L. Chen, F. Tung, J. He, M. Nawhal, and G. Mori, "LifeLong GAN: Continual learning for conditional image generation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 2759–2768.
[58] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.
[59] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Proc. 6th Indian Conf. Comput. Vis. Graph. Image Process.*, 2008, pp. 722–729.
[60] G. Patterson and J. Hays, "Sun attribute database: Discovering, annotating, and recognizing scene attributes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2751–2758.
[61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[62] S. Reed, Z. Akata, H. Lee, and B. Schiele, "Learning deep representations of fine-grained visual descriptions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 49–58.
[63] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 139–154.
[64] E. Schonfeld, S. Ebrahimi, S. Sinha, T. Darrell, and Z. Akata, "Generalized zero-and few-shot learning via aligned variational autoencoders," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8247–8255.
[65] S. Narayan, A. Gupta, F. S. Khan, C. G. Snoek, and L. Shao, "Latent embedding feedback and discriminative features for zero-shot classification," 2020. [Online]. Available: arXiv:2003.07833.

Kun Wei received the B.E. degree in electronic and information engineering from Xidian University, Xi'an, China, in 2017, where he is currently pursuing the Ph.D. degree with the School of Electronic Engineering.

His research interests lie primarily in computer vision and machine learning.

Cheng Deng (Senior Member, IEEE) received the B.E., M.S., and Ph.D. degrees in signal and information processing from Xidian University, Xi'an, China, in 2001, 2006, and 2009, respectively.

He is currently a Full Professor with the School of Electronic Engineering, Xidian University. He has authored and coauthored of more than 100 scientific articles at top venues, including IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, International Conference on Computer Vision and Pattern Recognition, International Conference on Machine Learning, Neural Information Processing Systems, International Joint Conference on Artificial Intelligence, and Association for Advancement of Artificial Intelligence. His research interests include computer vision, pattern recognition, and information hiding.

Xu Yang (Member, IEEE) received the B.E. and Ph.D. degrees in electronic and information engineering from Xidian University, Xi'an, China, in 2016 and 2021, respectively.

He is currently a Lecturer with the School of Electronic Engineering, Xidian University. His research interests lie primarily in computer vision and machine learning.

Dacheng Tao (Fellow, IEEE) is the President of the JD Explore Academy, Beijing, China, and a Senior Vice President of JD.com. He is also an Advisor and a Chief Scientist of the Digital Science Institute, University of Sydney, Sydney, NSW, Australia. He mainly applies statistics and mathematics to artificial intelligence and data science, and his research is detailed in one monograph and over 200 publications in prestigious journals and proceedings at leading conferences.

Dr. Tao received the 2015 Australian Scopus-Eureka Prize, the 2018 IEEE ICDM Research Contributions Award, and the 2021 IEEE Computer Society McCluskey Technical Achievement Award. He is a Fellow of the Australian Academy of Science, AAAS, and ACM.