MetaZSCIL: A Meta-Learning Approach for Generalized Zero-Shot Class Incremental Learning

Yanan Wu[1*], Tengfei Liang[1*], Songhe Feng[1†], Yi Jin[1†], Gengyu Lyu[2], Haojun Fei[3], Yang Wang[4]

[1]Beijing Key Laboratory of Traffic Data Analysis and Mining
School of Computer and Information Technology, Beijing Jiaotong University
[2]Faculty of Information Technology, Beijing University of Technology
[3]360 DigiTech, Inc
[4]Department of Computer Science and Software Engineering, Concordia University
{ynwu0510, tengfei.liang, shfeng, yjin}@bjtu.edu.cn, lyugengyu@bjut.edu.cn, zhangchulan-jk@360shuke.com, yang.wang@concordia.ca

Abstract

Generalized zero-shot learning (GZSL) aims to recognize samples whose categories may not have been seen at training. Standard GZSL cannot handle dynamic addition of new seen and unseen classes. In order to address this limitation, some recent attempts have been made to develop continual GZSL methods. However, these methods require end-users to continuously collect and annotate numerous seen class samples, which is unrealistic and hampers the applicability in the real-world. Accordingly, in this paper, we propose a more practical and challenging setting named Generalized Zero-Shot Class Incremental Learning (CI-GZSL). Our setting aims to incrementally learn unseen classes without any training samples, while recognizing all classes previously encountered. We further propose a bi-level meta-learning based method called MetaZSCIL to directly optimize the network to learn how to incrementally learn. Specifically, we sample sequential tasks from seen classes during the offline training to simulate the incremental learning process. For each task, the model is learned using a meta-objective such that it is capable to perform fast adaptation without forgetting. Note that our optimization can be flexibly equipped with most existing generative methods to tackle CI-GZSL. This work introduces a feature generative framework that leverages visual feature distribution alignment to produce replayed samples of previously seen classes to reduce catastrophic forgetting. Extensive experiments conducted on five widely used benchmarks demonstrate the superiority of our proposed method.

Introduction

Generalized Zero-Shot Learning (GZSL) aims to tackle the unseen classes recognition problem by transferring semantic knowledge of seen classes to unseen ones (Liu et al. 2021; Feng et al. 2022). Typically, these models are offline trained on a set of predefined seen classes and then deployed to target applications with fixed parameters. Such systems are not flexible enough since they cannot sequentially learn and accumulate the knowledge of new classes that might emerge after deployment. In contrast, humans are able to learn new concepts incrementally throughout their lifetime.

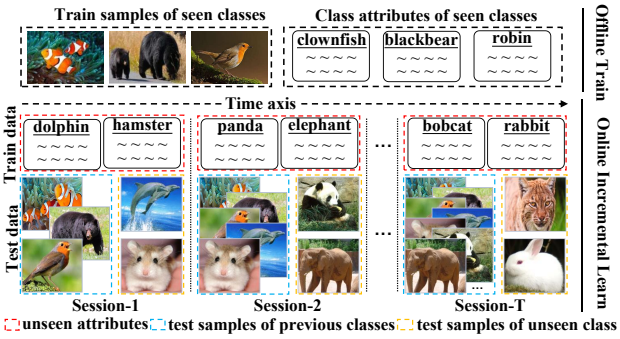


Figure 1: Illustration of our CI-GZSL setting. During the offline training, we learn an initial model based on samples and attributes of seen classes. During each subsequent online incremental learning, we are given the attributes of some unseen classes without any training examples (*i.e.*, zero-shot). Our goal is to update the model in each incremental session, so that the model can recognize all classes (including those during the offline stage) encountered so far.

Recently, sporadic efforts (Wei et al. 2020; Ghosh 2021) have been made towards designing models that can dynamically adapt and generalize on the addition of new seen and unseen classes. The aforementioned works formulate this setting as *continual GZSL*. Lifelong ZSL (Wei et al. 2020) marks the first attempt to address the problem of continual GZSL. It considers each dataset as an incremental session to accumulate the knowledge from multiple datasets during training, then separately recognizes unseen classes of all encountered datasets based on task-ids. However, this approach requires task-level supervision at test time, which limits its applicability in realistic scenarios. (Skorokhodov and Elhoseiny 2021) proposes a class normalization-based approach for the continual GZSL problem, in which the dataset is divided into multiple subsets and the model encounters each of these subsets in an incremental fashion over time. The setting assumes all previously encountered ses-

*The authors contributed equally to this work.
†Corresponding Author

sions as seen and future sessions as unseen classes. It requires the total number of classes or sessions to be known beforehand. But this fundamentally violates the concept of continual learning. The latest works (Kuchibhotla et al. 2022; Gautam et al. 2022) reformulate continual GZSL setting where each session has an exclusive set of seen and unseen classes and the model can accommodate any number of sessions over time. However, this setting requires large-scale annotated samples for new seen classes in each incremental session. This is unrealistic in practice since it requires end-users to continually provide numerous training samples.

We argue that a practical continual GZSL model should incrementally learn new classes without any training samples, while being able to identify all classes previously encountered. It should not require end-users to collect any annotated samples, since fine-grained annotation is laborious and often requires expert domain knowledge. In this paper, we propose a more challenging setting named *Generalized Zero-Shot Class Incremental Learning* (CI-GZSL). CI-GZSL consists of an offline training stage and an online incremental learning stage, as shown in Figure 1. In the offline training stage, CI-GZSL requires the model to learn the initialization weights based on annotated samples of *seen* classes. In the online incremental learning (*i.e.*, evaluation) stage, the model will encounter a few *unseen* classes at each incremental session, where these *unseen* classes only have their corresponding attribute descriptors without any annotated samples. Besides, the model cannot store training samples from *seen* classes during evaluation, due to data privacy and limited storage of the deployment environment. The evaluation protocol is defined such that after learning the *unseen* classes in each incremental session, the model is evaluated on all encountered classes (including *seen* classes). CI-GZSL is challenging due to two main reasons, namely catastrophic forgetting of seen classes and adaptation ability of unseen classes in incremental learning.

To overcome such issues, we propose a meta-learning based approach called MetaZSCIL for CI-GZSL. MetaZSCIL directly formulates forgetting alleviation and incremental adaptation as the optimization objective. Specifically, in the offline training stage, we employ a sequential task sampling scheme to mimic the incremental learning process of the evaluation stage. For each incremental session, the model first performs fast learning based on seen classes via a few gradient updates. Then the meta-objective is defined by evaluating the learned model on the test images of previous encountered classes (test forgetting) and current unseen classes (test adaptation). The goal of our meta-learning is to learn a model initialization such that it can sequentially adapt to unseen classes and is less prone to catastrophic forgetting. The above optimization is built upon a feature-generative network, in which we further propose a visual distribution alignment-based replay strategy to transfer previously learned knowledge to the current incremental session. The major contributions are summarized as follows:

- We propose a practical yet more challenging CI-GZSL setting that is user-friendly and more realistic for applications. In this setting, the user can easily expand the model capability to recognize both previously seen classes and user-specific unseen classes encountered over time by only providing the attribute descriptors of unseen classes.
- We design a sequential task sampling scheme to mimic the incremental learning process at evaluation, and accordingly propose a bi-level optimization MetaZSCIL based on meta-learning. Our method explicitly trains the network to facilitate fast learning of new unseen classes and is robust to forgetting under online updates.
- We propose a feature generative framework that replays samples closer to true distribution of the original data via a visual distribution alignment loss. This allows our model to effectively accumulate learned knowledge from previous sessions and enable generalization.
- Extensive experimental results on five benchmarks, *i.e.*, AWA1, AWA2, CUB, SUN and aPY, clearly demonstrate the advantages of the proposed MetaZSCIL over other baseline and current state-of-the-art methods.

Related Work

Generalized Zero-Shot Learning (GZSL). Existing GZSL approaches can be broadly classified into embedding-based methods (Cacheux, Borgne, and Crucianu 2019) and generative methods (Kong et al. 2022). Traditional embedding-based methods aim to project visual and/or semantic features onto a common embedding space and use a nearest neighbor-based classifier to classify visual samples. Feature generation-based methods are proposed to synthesize unseen visual features, thus converting the GZSL problem into a supervised classification problem. Although these methods have shown promising performance, they cannot continually learn from sequential streaming data. To this end, some recent works (Verma et al. 2021) have drawn increasing attention towards continual GZSL. (Wei et al. 2022) utilizes generative replay and knowledge distillation to facilitate the sequential accumulation of knowledge to improve new classes recognition. (Kuchibhotla et al. 2022) proposes a feature-generative framework based on bi-directional incremental alignment to avoid catastrophic forgetting and enable continual generalization. However, these methods only consider the incremental steps at test time, which leads to sub-optimal performance due to the misalignment between their optimization objectives and evaluation protocol.

Class-Incremental Learning (CIL). Class-incremental learning aims at continuously updating a trained model with new classes without forgetting previously learned old ones (Yan, Xie, and He 2021; Castro et al. 2018; Lyu et al. 2022). To mitigate the forgetting of the old classes, CIL methods (Hu et al. 2021; Kang, Park, and Han 2022) typically adopt the knowledge distillation technique, where external memory is often used for storing old class samples to compute the distillation loss. iCaRL (Rebuffi et al. 2017) maintains an "episodic memory" of the samples and incrementally learns the nearest-neighbor classifier for the new classes. NCM (Hou et al. 2019) introduces cosine normalization to balance between the classifier for old and novel data. In contrast to these CIL works, we focus on the more challenging CI-GZSL problem, where there are only attribute descriptors

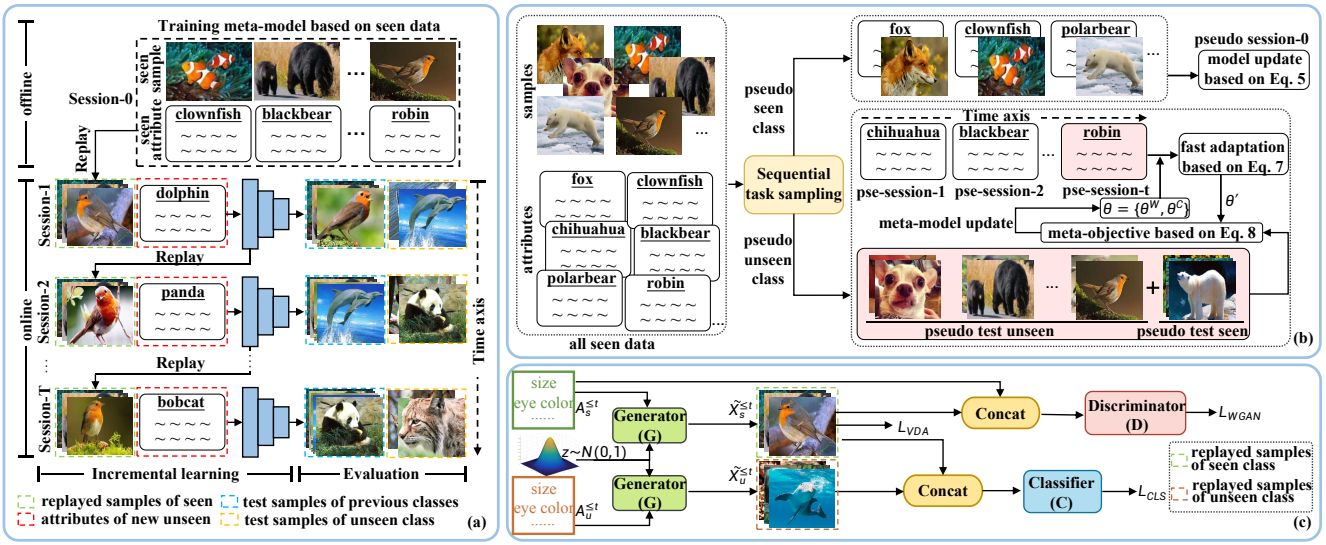


Figure 2: Overview of the proposed MetaZSCIL. (a) Our offline training and online incremental learning process in the setting of CI-GZSL. (b) Our optimization strategy based on meta-learning during offline training. (c) Our generative network based on visual distribution alignment to alleviate catastrophic forgetting and generalize knowledge of seen to unseen classes.

without any training samples for each new class. Rather than directly storing samples of old classes, our method uses a replay mechanism to accumulate the knowledge of old classes.

Meta-Learning. Existing meta-learning methods include model-based (Santoro et al. 2016; Wu et al. 2021), metric-based (Koch et al. 2015; Vinyals et al. 2016) and optimization-based (Finn, Abbeel, and Levine 2017; Chi et al. 2022, 2021). Our proposed method builds upon the most popular model agnostic meta-learning (MAML) algorithms (Finn, Abbeel, and Levine 2017). MAML learns the model using a nested optimization, where the inner loop performs task-level optimization, while the outer loop performs a meta-level model update via meta-objective. The goal of MAML is to learn a model initialization such that it can quickly adapt to new tasks. In this paper, we advance the MAML to automatically learn the optimal trade-off between two competing factors, namely adapting to new classes knowledge and retaining the knowledge of old classes.

The Proposed Method

Problem Definition. CI-GZSL aims to incrementally learn unseen classes with only attribute descriptors by accumulating past knowledge. Let subscripts s and u denote seen and unseen classes respectively. We define a sequence of incremental sessions $\{\mathcal{D}^0, \mathcal{D}^1, \cdots, \mathcal{D}^T\}$ and their corresponding class set $\mathcal{C}^t$ at session t ($t = 0, 1, \cdots, T$). Note that the class sets are disjoint among different sessions ($\mathcal{C}^i \cap \mathcal{C}^j = \varnothing (i \neq j)$). Only the classes in the first session $\mathcal{D}^0$ contains large-scale training samples, *i.e.*, $\mathcal{D}^0 = \{$(training samples of seen classes $\mathcal{X}^0_{s_{tr}}$, their class label $\mathcal{Y}^0_{s_{tr}}$, class attribute $\mathcal{A}^0$)$\}$. We refer $\mathcal{C}^0$ as *seen* classes. The offline training is performed using these *seen* classes to learn a model initialization in $\mathcal{D}^0$. Once the offline training stage is done, CI-GZSL requires the model to adapt to *unseen* classes in subsequent incremental sessions ($\mathcal{D}^t, t{>}0$). Each subsequent incremental session only requires the attribute of *unseen* classes in $\mathcal{C}^t$. After learning on $\mathcal{D}^t$, the model is evaluated on test images of all encountered classes so far, *i.e.*, $\mathcal{C}^0 \cup \mathcal{C}^1 \cdots \cup \mathcal{C}^t$. We assume our model has access to class attributes of all encountered classes so far. Let $\mathcal{A}^t$ be the union of class attributes of seen ($\mathcal{A}^t_s$) and unseen ($\mathcal{A}^t_u$) classes encountered so far. So $\mathcal{D}^t_{tr} = \{$(class attribute $\mathcal{A}^t$)$\}$ and $\mathcal{D}^t_{te} = \{$(test samples of seen and unseen classes encountered so far ($\mathcal{X}^t_{s_{te}}$ and $\mathcal{X}^t_{u_{te}}$), their class labels ($\mathcal{Y}^t_{s_{te}}$ and $\mathcal{Y}^t_{u_{te}}$), class attribute $\mathcal{A}^t$)$\}$.

CI-GZSL is motivated by challenges in real-world applications. Consider the scenario of a company that provides image classification models to users. The company can do an offline training with annotated samples of seen classes to get the model. After the model is deployed to a user, the user may want to incrementally adapt the model to recognize new object classes of interest to this specific user. For the end-user, it is not practical to access the labeled seen samples from the company or to provide annotated samples of new (unseen) classes. It is much desirable if the user can simply provide the semantic information (e.g. attribute descriptors) of the new objects. Our proposed CI-GZSL setting can be used to solve the above practical scenario, which recognizes both the seen classes and the user-specific unseen classes encountered over a period of time using only attribute descriptors of unseen classes.

Feature Generation-based GAN Classifier

Revisiting f-CLSWGAN. In this paper, we use f-CLSWGAN (Xian et al. 2018b) as our backbone to learn the $semantic \rightarrow visual$ mapping. f-CLSWGAN is composed of a Wasserstein Generative Adversarial Network (WGAN) and a classifier. WGAN consists of a generator G and a discriminator network D that competes in a two-player mini-

max game. The generator $G(z, a)$ synthesizes a visual feature $\tilde{x}$ with a random input noise z, whereas the discriminator $D(x, a)$ takes a visual feature x (real or synthetic) and outputs a real value indicating the degree of realness or fakeness. Both G and D are conditioned on the class attribute a. Learning of G and D is done by optimizing WGAN loss:

$$\mathcal{L}_{WGAN} = \mathbb{E}[D(x,a)] - \mathbb{E}[D(\tilde{x},a)] - \lambda\mathbb{E}[(||\nabla_{\hat{x}} D(\hat{x},a)||_2 - 1)^2], \quad (1)$$

where $\tilde{x} = G(z, a)$, $\hat{x} = \epsilon x + (1-\epsilon)\tilde{x}$ with $\epsilon \sim U(0, 1)$, and λ is the coefficient of gradient penalty that enforces the gradient of D to have a unit norm along the straight line between pairs of real and generated points (Gulrajani et al. 2017). Through this adversarial training, WGAN learns to generate visual features similar to the real visual features. Then a standard softmax classifier is trained with seen samples and synthesized unseen samples to distinguish different classes. The classification loss is defined as follows:

$$\mathcal{L}_{CLS} = -\mathbb{E}_{\tilde{x}\sim p_{\tilde{x}}}[\log P(y|\tilde{x};\theta^C)], \quad (2)$$

where y is the class label of $\tilde{x}$ and $P(y|\tilde{x};\theta^C)$ denotes the probability of $\tilde{x}$ corresponding to its true class label y.

Generative Replay. We work in a setting where new object classes are continually learned over time, and samples from previous sessions are not accessible during the current session. This results in catastrophic forgetting. In order to retain learned knowledge, we use generative replay to synthesize visual features of previously seen classes. Given the relatively weak constraint of the above generation process, it is possible to produce visual features that are too far from the true distribution present in the training set, resulting in ineffective multi-class classifier training (Gong, Yuan, and Bao 2021, 2022a). To this end, we propose a visual distribution alignment loss ($\mathcal{L}_{VDA}$) consisting of class consistency ($\mathcal{L}_{CC}$) loss and sample diversity ($\mathcal{L}_{SD}$) loss. The former guarantees that the generated visual features contain representative features of the classes, while the latter encourages the generator to learn more discriminative class-relevant features for better generalization, as shown in Figure 2 (c).

$$\mathcal{L}_{CC} = \frac{1}{|\mathcal{C}^0|}\sum_{c=1}^{|\mathcal{C}^0|}(\overline{x}_c - \tilde{x}_c)^2, \quad (3)$$

$$\mathcal{L}_{SD} = \frac{1}{|\mathcal{C}^0|}\sum_{c=1}^{|\mathcal{C}^0|}\sum_{i=1}^{M}\sum_{\substack{j=1,\\ j\neq i}}^{M}\text{sim}(\tilde{x}_c^i, \tilde{x}_c^j), \quad (4)$$

where $\overline{x}_c$ / $\tilde{x}_c$ denotes visual prototype of seen class c, which is computed by averaging all real / replayed visual features from this class. $\mathcal{L}_{CC}$ is used to narrow the distance between the above two prototypes to achieve intra-class compactness and inter-class separability, and $|\mathcal{C}^0|$ is the number of all seen classes. Besides, $\text{sim}(\tilde{x}_c^i, \tilde{x}_c^j)$ denotes cosine similarity between samples i and j, where $\tilde{x}_c^i$ stands for the i^{th} generated feature of class c, and M is the number of replayed samples. $\mathcal{L}_{SD}$ tries to minimize visual similarity among generated features of each seen class to improve intra-class sample diversity. Note that $\mathcal{L}_{VDA}$ optimizes the parameters of the generator G. Our final objective function is defined as

$$\mathcal{L} = \min_G \max_D (\mathcal{L}_{WGAN} + \mathcal{L}_{VDA}) + \mathcal{L}_{CLS}. \quad (5)$$

At the test time, given the attributes of unseen classes from the current incremental session, we combine it with Gaussian noise and generate corresponding synthetic visual features $\tilde{\mathcal{X}}^t$. In order to accumulate previously learned knowledge, we further generate replay features $\tilde{\mathcal{X}}^{t'}$ of previously encountered seen classes by the generator network. A combination of synthetic features of previous all classes and unseen classes from the current session act as input data to train a new softmax classifier (Liu et al. 2018). The classifier minimizes the negative log-likelihood loss as follows:

$$\min_{\theta^C} -\frac{1}{|\mathcal{T}|}\sum_{(\tilde{x},y)\in\mathcal{T}}\log P(y|\tilde{x};\theta^C), \quad (6)$$

where θ^C is the weight matrix of a fully connected layer that projects the visual feature $\tilde{x}$ to C probabilities, with C being the number of all classes encountered so far. Here $\mathcal{T} = (\tilde{\mathcal{X}}^t, \mathcal{Y}^t) \cup (\tilde{\mathcal{X}}^{t'}, \mathcal{Y}^{t'})$, where $\mathcal{Y}^t$ and $\mathcal{Y}^{t'}$ are the class labels corresponding to $\tilde{\mathcal{X}}^t$ and $\tilde{\mathcal{X}}^{t'}$, respectively. Finally, the prediction function of the test data $x \in \mathcal{X}_{s_{te}}^t \cup \mathcal{X}_{u_{te}}^t$ is $\arg\max P(y|x;\theta^C)$, which is used to generate predicted labels and evaluate the performance of the method.

Algorithm 1: The optimization procedure of MetaZSCIL

Require: α, β, γ: learning rates
Require: $\mathcal{D}^0$: training set of *seen* classes
randomly initialize parameters θ
while not converged **do**
  $\mathcal{D}^s = \{(\mathcal{D}_{p_{tr}}^j, \mathcal{D}_{p_{te}}^j)\}_{j=0}^N$ ▷ sample a pseudo sequence
  $\mathcal{D}_{p_{te}} = \varnothing$ ▷ empty cumulative pseudo test set
  $\mathcal{A}_p = \varnothing$ ▷ empty cumulative pseudo class attribute
  $\theta^W \leftarrow \theta^W - \gamma\nabla_{\theta^W}\mathcal{L}_{WGAN}(\mathcal{X}_{ps_{tr}}^0, \mathcal{Y}_{ps_{tr}}^0, \mathcal{A}_p^0; \theta^W)$
    ▷update parameters using pseudo seen classes
  $\mathcal{D}_{p_{te}} = \mathcal{D}_{p_{te}} \cup \mathcal{D}_{p_{te}}^0$; $\mathcal{A}_p = \mathcal{A}_p \cup \mathcal{A}_p^0$
  for $j = 1, \cdots, N$ **do**
    $\theta^{W'} = \theta^W - \alpha\nabla_{\theta^W}\mathcal{L}_{WGAN}(\mathcal{A}_p; \theta^W)$
    ▷ compute adapted params with all previous classes
    $\mathcal{D}_{p_{te}} = \mathcal{D}_{p_{te}} \cup \mathcal{D}_{p_{te}}^j$; $\mathcal{A}_p = \mathcal{A}_p \cup \mathcal{A}_p^j$
      ▷accumulate the test set and class attribute
    $\theta \leftarrow \theta - \beta\nabla_\theta \sum_{(\mathcal{X}_{p_{te}}, \mathcal{Y}_{p_{te}})\in\mathcal{D}_{p_{te}}} \mathcal{L}(\mathcal{X}_{p_{te}}, \mathcal{Y}_{p_{te}}, \mathcal{A}_p; \theta^{W'}, \theta^C)$
      ▷ update meta-model θ to adapt to new session
  end for
end while

Learning to Incrementally Learn

The optimization scheme of our method is inspired by MAML (Finn, Abbeel, and Levine 2017) for few-shot learning. During the meta-training stage, MAML learns from a set of tasks, in which each task is constructed as a few-shot learning problem to mimic the scenario during meta-testing. In CI-GZSL, we regard the online incremental stage

as the "meta-testing stage" in MAML. The online incremental stage adapts the model to a sequence of incremental sessions, where each session involves several unseen classes with only attribute descriptors. As Figure 2 (a) shows, we simulate this scenario during the offline training stage and use a meta-learning algorithm to learn a model initialization from the seen classes. The high-level idea of our method is to use seen classes to mimic the incremental learning scenario that the model will encounter during the online incremental learning, so that the model is learned in a way that it can effectively adapt to unseen classes during evaluation.

Sequential Task Sampling. In the offline training, we mimic the online incremental process to learn a model initialization using the *seen* classes. Specifically, at each task, we first randomly separate the *seen* classes as *pseudo seen classes* and *pseudo unseen classes* without overlapping. Next we sample a sequence of $T+1$ sessions, $\mathcal{D}^s = \{(\mathcal{D}^j_{tr}, \mathcal{D}^j_{te})\}^T_{j=0}$, where $\mathcal{D}^j_{tr}$ and $\mathcal{D}^j_{te}$ are the training and test set for the j^{th} session. Note that we set $(\mathcal{D}^0_{tr}, \mathcal{D}^0_{te})$ as the *pseudo seen* set with more classes and training samples, *i.e.*, $\mathcal{D}^0_{tr}$= {(training samples of pseudo seen classes $\mathcal{X}^0_{ps_{tr}}$, their class labels $\mathcal{Y}^0_{ps_{tr}}$, class attributes $\mathcal{A}^0_p$)}, $\mathcal{D}^0_{te}$= {(test samples of pseudo seen classes $\mathcal{X}^0_{ps_{te}}$, their class labels $\mathcal{Y}^0_{ps_{te}}$, class attributes $\mathcal{A}^0_p$)}. The subsequent sessions (*e.g.* $j>0$) follow the online incremental learning setting as evaluation, such as $\mathcal{D}^j_{tr}$= {(the attributes of *pseudo unseen class* from current session j ($\mathcal{A}^j_p$))}, $\mathcal{D}^j_{te}$= {(test samples of pseudo seen classes and unseen classes encountered so far ($\mathcal{X}^j_{ps_{te}}$ and $\mathcal{X}^j_{pu_{te}}$), their class labels ($\mathcal{Y}^j_{ps_{te}}$ and $\mathcal{Y}^j_{pu_{te}}$), class attributes ($\mathcal{A}^j_p$))}.

Meta-Training. For each sampled sequence $\mathcal{D}^s$, we propose a bi-level optimization based on MAML to directly formulate incrementally adapting without forgetting as the meta-objective. We denote $\theta = \{\theta^W, \theta^C\}$ as the parameters for the whole network, where θ^W and θ^C denote the parameters for WGAN and classifier. The meta-learning procedure is illustrated in Alg. 1 and Figure 2 (b). At the beginning of training, we respectively define an empty cumulative pseudo test set $\mathcal{D}_{p_{te}}$ and class attribute set $\mathcal{A}_p$ to store the test samples and class attributes from previous sessions. After that, we conduct supervised training of θ^W on the pseudo seen classes ($j = 0$) using the Wasserstein GAN loss. In each subsequent session ($j>0$), we perform fast adaptation to all classes previously encountered and update θ^W via a few gradient steps for the inner-loop task-level optimization:

$$\theta^{W'} = \theta^W - \alpha \nabla_{\theta^W} \mathcal{L}_{WGAN}(\mathcal{A}_p; \theta^W), \tag{7}$$

where $\mathcal{A}_p$ is the attribute set of all pseudo seen and unseen classes from the beginning to the session $j-1$. The loss $\mathcal{L}_{WGAN}(\cdot)$ means that WGAN learns a better mapping for the class attributes. In other words, generative network G learns to generate visual features similar to the real features of all previous classes (Gong, Yuan, and Bao 2022b). This alleviates catastrophic forgetting of seen classes and helps new unseen classes to synthesize visual features.

The above adaptation process mimics how the model learns unseen classes at test time. Ideally, we would like the adapted parameters to perform well on the classes from the previous and current sessions. The test sets from previous sessions reflect whether the updated model is robust to the catastrophic forgetting, while the test set of the current session validates model adaptation to new unseen classes (Chi et al. 2020). Thus, we append $\mathcal{D}^j_{p_{te}}$ to $\mathcal{D}_{p_{te}}$. To predict all classes encountered so far, we also append $\mathcal{A}^j_p$ to $\mathcal{A}_p$ since our classifier is obtained based on a linear mapping over their class attributes. Accordingly, the meta-objective is defined as follows for the outer-loop meta-level optimization:

$$\min_{\theta^W, \theta^C} \sum\nolimits_{(\mathcal{X}_{p_{te}}, \mathcal{Y}_{p_{te}}) \in \mathcal{D}_{p_{te}}} \mathcal{L}(\mathcal{X}_{p_{te}}, \mathcal{Y}_{p_{te}}, \mathcal{A}_p; \theta^{W'}, \theta^C). \tag{8}$$

Note that $\mathcal{L}(\cdot)$ is a function of $\theta^{W'}$ but the optimization is performed on θ^W. We then minimize the meta-objective in Eq. 8 using gradient decent, as shown in Line 14 of Alg. 1. When all $N+1$ sessions in a task are done, $\mathcal{D}_{p_{te}}$ and $\mathcal{A}_p$ are reset to empty as they can be dynamically extended to any length in terms of different tasks.

Meta-Testing. The meta parameter θ^C is learned to perform fast adaptation via synthesized visual features of new unseen classes. And θ^W is trained to facilitate the learning process without forgetting based on the replayed samples of all classes previously encountered. During the online learning, we perform Line 10 of Alg. 1 to accumulate historical knowledge to synthesize discriminative visual features for new unseen classes. The procedure in Alg. 1 matches the evaluation protocol: at each incremental session, the model is evaluated on all encountered classes after training on the current session. Our meta-objective optimizes the model towards what it is supposed to do at evaluation.

Experiments

Datasets and Setup

Dataset. We conduct experiments on five widely used ZSL datasets: Animals with Attributes 1&2 (**AWA1** (Lampert, Nickisch, and Harmeling 2013) & **AWA2** (Xian et al. 2018a)), UCSD Birds-200-2011 (**CUB** (Wah et al. 2011)), Scene Recognition (**SUN** (Patterson and Hays 2012)), and Attributes Pascal and Yahoo (**APY**(Farhadi et al. 2009)). AWA1 and AWA2 share the same 50 animal classes with 85-dimensions attributes. AWA1 includes 30,475 images and AWA2 consists of 37,322 images. They are split into 40 seen classes and 10 unseen classes; CUB contains 11,788 images of 200 bird species in which 150 classes are treated as seen and 50 classes are unseen; SUN consists of 14,340 fine-grained images from 717 classes, including 645 seen classes and 72 unseen classes. In aPY, there are 15,339 images belonging to 32 classes, and 20 of these classes are treated as seen and 12 are unseen.

Session-wise Data Split. In our CI-GZSL setting, all seen classes are used for offline training (*i.e.*, session 0). The unseen classes are dynamically added in each subsequent session (*i.e*, session $1\sim T$) during the online incremental learning stage. In this work, we focus on the data split of unseen classes. Specifically, AWA1 and AWA2 datasets are divided into 5 incremental sessions. In each session, 2 new unseen

Methods	**AWA1** (Session Number)					Average	Final	**AWA2** (Session Number)					Average	Final
	1	2	3	4	5	Acc	Impro.	1	2	3	4	5	Acc	Impro.
f-CLSWGAN	67.97	62.07	53.98	45.45	42.08	54.31	+24.91	74.63	67.02	59.29	53.43	48.40	60.55	+19.10
CADA-VAE	85.59	76.79	70.07	67.75	62.74	72.59	+4.25	88.78	82.30	74.14	70.87	63.30	75.88	+4.20
CN-CZSL	**88.20**	74.58	70.55	68.93	65.06	73.46	+1.93	**90.39**	74.75	70.35	68.63	65.50	73.92	+2.00
Online-CGZSL	77.61	67.48	65.36	63.28	61.14	66.97	+5.85	71.82	61.46	62.76	60.62	57.78	62.89	+9.72
MetaZSCIL(ours)	86.31	**81.45**	**73.99**	**71.61**	**66.99**	**76.07**		86.19	**85.16**	**80.15**	**73.83**	**67.50**	**78.57**	

Methods	**aPY** (Session Number)				Average	Final	**SUN** (Session Number)								Average	Final
	1	2	3	4	Acc	Impro.	1	2	3	4	5	6	7	8	Acc	Impro.
f-CLSWGAN	40.13	28.53	21.93	21.62	28.05	+20.67	29.96	27.90	25.81	24.58	24.78	24.33	25.98	25.12	26.06	+17.09
CADA-VAE	64.64	51.79	46.92	40.48	50.96	+1.81	45.74	45.72	43.89	43.68	42.45	40.78	40.39	39.55	42.78	+2.66
CN-CZSL	60.86	46.41	44.24	41.13	48.16	+1.16	**50.48**	49.18	47.98	45.91	42.59	42.75	40.88	38.39	44.77	+3.82
Online-CGZSL	46.56	38.38	39.81	33.28	39.51	+9.01	41.06	37.31	38.44	38.14	38.85	37.20	38.46	37.81	38.40	+4.40
MetaZSCIL(ours)	**69.35**	**55.56**	**47.92**	**42.29**	**53.78**		50.11	**50.31**	**48.72**	**46.98**	**45.94**	**44.55**	**43.65**	**42.21**	**46.56**	

Methods	**CUB** (Session Number)										Average	Final
	1	2	3	4	5	6	7	8	9	10	Acc	Impro.
f-CLSWGAN	33.44	30.52	24.53	22.62	19.91	17.49	15.77	14.86	13.90	11.80	20.48	+34.18
CADA-VAE	**63.03**	59.10	51.59	47.57	46.24	44.65	41.55	40.04	39.92	39.46	47.32	+6.52
CN-CZSL	39.55	51.58	49.88	48.12	48.23	49.89	48.79	47.82	45.90	44.21	47.40	+1.77
Online-CGZSL	42.39	48.42	43.36	41.92	42.30	43.92	43.09	41.81	40.26	38.91	42.64	+7.07
MetaZSCIL(ours)	40.43	**62.31**	**55.94**	**54.40**	**53.53**	**53.05**	**50.35**	**49.39**	**46.68**	**45.98**	**51.21**	

Table 1: Performance (in %) comparisons with the state-of-the-art methods on AWA1, AWA2, aPY, SUN, CUB datasets. The results of other methods are obtained by running their released codes under the CI-GZSL setting.

classes are added. CUB dataset is divided into 10 sessions, where 5 unseen classes are added in each incremental session. SUN dataset is divided into 8 incremental sessions with 9 unseen classes per session. The aPY dataset consists of 4 incremental sessions with 3 unseen classes in each session.

Sequential Task Sampling. In the offline training stage, we use the training set of *seen* classes to sample sequential tasks. We first split them into non-overlapping *pseudo seen* and *pseudo unseen* classes (30/10 for AWA1 and AWA2, 100/50 for CUB, 573/72 for SUN, 12/8 for aPY). For each sequence task, we randomly sample *pseudo seen* classes first (pre-training WGAN for generative replay in subsequent incremental sessions), followed by T incremental sessions with *pseudo unseen* classes. Both the session number and the number of unseen classes in each incremental session are consistent with the online incremental learning scenario.

Evaluation Metrics. We adopt three widely used zero-shot metrics to evaluate each comparison method, including the top-1 accuracy on seen classes (S), unseen classes (U) and their harmonic mean (defined as $H = 2 \times S \times U/(S + U)$). Besides, we report the average of all sessions such as mean seen accuracy (mSA), mean unseen accuracy (mUA) and mean harmonic value(mH). Note that our evaluation protocol is more practical and different from existing setting (Kuchibhotla et al. 2022; Gautam et al. 2022). In previous continual GZSL setting, during the online learning stage, the evaluation is performed only at the end of all sessions. However, in our CI-GZSL setting, the model is evaluated at each session during the online learning stage.

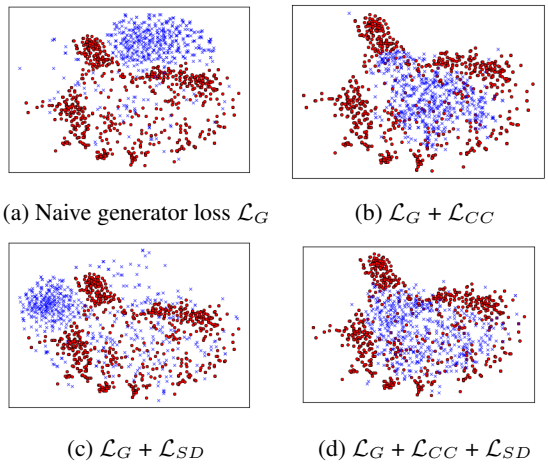
Implementation Details. Our networks are optimized by the Adam optimizer with β_1 = 0.5, β_2 = 0.999, and initial learning rate 0.0001 in both offline training and online incremental learning stages. The penalty coefficient λ is set to 10. The input noise in the generator has the same dimension as the corresponding attributes. We set mini-batch to 512 for AWA1 and AWA2, 64 for CUB, SUN and aPY. In the meta-training (offline) stage, we first perform supervised training on *pseudo seen* classes for 30 epochs, followed by 5 inner and 1 outer gradient updates for adapting new *pseudo unseen* classes without forgetting. In the meta-testing (online) stage, we directly perform 10 gradient updates to fast adapt *unseen* classes of each incremental session.

Experiment Results

Main Results. Since this paper considers a new problem setting, there is no prior work that we can directly compare. Nevertheless, we compare with state-of-the-art methods on GZSL by running their codes under our CI-GZSL setting, including classic GZSL baseline f-CLSWGAN (Xian et al. 2018b), CADA-VAE (Schonfeld et al. 2019), ans recent continual GZSL models CN-CZSL (Skorokhodov and Elhoseiny 2021), Online-CGZSL (Kuchibhotla et al. 2022). We report the harmonic mean between seen and unseen classes for each incremental session and the average of all incremental sessions. We also show relative improvement for the final session. As shown in Table 1, the proposed method outperforms all comparison methods on all five datasets among most incremental sessions. Specifically, our MetaZSCIL surpasses the most recent method online-CGZSL by 5.85%, 9.72%, 9.01%, 4.40 and 7.07% on AWA1, AWA2,

Methods	**AWA1** (Session Number)															Average Acc		
	1			2			3			4			5					
	S	U	H	S	U	H	S	U	H	S	U	H	S	U	H	mSA	mUA	mH
+ Baseline	54.61	90.01	67.98	51.55	78.00	62.07	42.97	72.57	53.98	34.82	65.42	45.45	40.97	43.24	42.07	44.98	69.85	54.31
+$\mathcal{L}_{CC}$	74.25	93.08	82.60	74.82	81.55	78.04	73.04	76.50	74.73	67.72	70.26	68.97	67.67	61.76	64.58	71.50	76.63	73.78
+$\mathcal{L}_{SD}$	75.32	91.66	82.69	75.62	82.76	79.03	73.82	76.95	75.35	69.17	71.80	70.46	66.99	64.15	65.54	72.18	77.47	74.62
+ Meta-learning	83.63	89.18	86.31	82.58	80.35	81.45	75.94	72.14	73.99	73.67	69.66	71.61	72.16	62.51	66.99	77.60	74.77	76.07

Table 2: Ablation study of various components of our MetaZSCIL on the AWA1 dataset. For each incremental session and the average of all incremental sessions, we report their seen accuracy (S), unseen accuracy (U) and their harmonic mean (H).



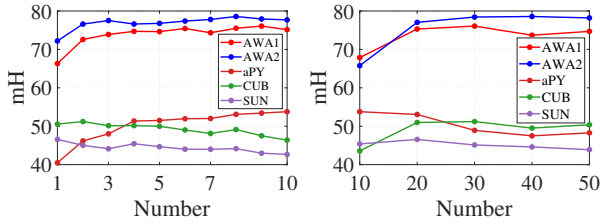
(a) Naive generator loss $\mathcal{L}_G$ (b) $\mathcal{L}_G$ + $\mathcal{L}_{CC}$

(c) $\mathcal{L}_G$ + $\mathcal{L}_{SD}$ (d) $\mathcal{L}_G$ + $\mathcal{L}_{CC}$ + $\mathcal{L}_{SD}$

Figure 3: The t-SNE visual results of sample distributions on the AWA1 dataset. The red and blue respectively represent the data distribution of the real and replayed samples.

aPY, SUN and CUB datasets for final accuracy. We also outperform the strong CN-CZSL method by 1.93%, 2.00%, 1.16%, 3.82% and 1.77%. This shows that learning to incrementally learn during the offline training stage is crucial for alleviating forgetting and promoting future learning.

Ablation Studies. To evaluate the effectiveness of each component in our proposed framework, we conduct ablation studies on the AWA1 dataset as shown in Table 2. The baseline model is the sequential version of f-CLSWGAN. Considering the relatively weak constraint of generator in the baseline model, we gradually employ class consistency loss ($\mathcal{L}_{CC}$) and sample diversity loss ($\mathcal{L}_{SD}$) which result in 19.47% and 0.84% higher harmonic mean. To intuitively understand their effectiveness, we randomly select one class and visualize the data distribution of the real and replayed samples under different losses. The results are shown in Figure 3. We can observe that the distributions with $\mathcal{L}_{CC}$ (Figure 3 (b)) are closer to real visual center while that with $\mathcal{L}_{SD}$ (Figure 3 (c)) exhibit the diversity of samples, which are consistent with their functions. The closest approximation to the true distribution is achieved when $\mathcal{L}_{CC}$ and $\mathcal{L}_{SD}$ are both applied, as shown in Figure 3 (d). Besides, our meta-learning optimization strategy further improves the harmonic mean of seen and unseen classes to 76.07% in the last row of Table 2. Particularly, it greatly boosts the seen class accuracies per incremental session and improves robustness to forgetting.



(a) Sampling sequence tasks (b) Replay samples per unseen

Figure 4: We compare the Mean Harmonic accuracy (mH) with different hyper-parameter on all datasets.

Hyper-parameter Sensitivity. We study the performance of the proposed method given different parameter settings. We first vary the number S of sampling sequential tasks in meta-training. According to Figure 4 (a), we find that more serialization tasks are beneficial for coarse-grained datasets, but are prone to overfitting for fine-grained datasets. Empirically, the optimal value of S is set to 9 in the AWA1, AWA2 and aPY datasets, and 2 in the CUB and SUN datasets. Then we evaluate the performance of different replayed samples number per unseen classes in Figure 4 (b). We observe that AWA1 and AWA2 perform well when the number of replayed samples is more than 20. We use a replay of 30 for both datasets in this work. For aPY, CUB and SUN datasets, we replay only 20 samples per unseen class.

Conclusion

This paper has proposed a practical and challenging setting called CI-GZSL. The goal is to incrementally learn unseen classes without any training samples, while keeping the knowledge of previously learned classes. To fast adapt to new classes without forgetting old classes during evaluation, we further propose a bi-level meta-learning-based optimization strategy to directly optimize the network to learn how to incrementally learn. Furthermore, our learning network is designed as a visual distribution alignment-based generative framework that can replay class-related discriminative features with robustness to forgetting. Extensive experiments demonstrate the superiority of our proposed method.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities (No. 2022JBZY019), the NSERC Discovery Grant, the National Natural Science Foundation of China (No. 61872032, No. 61972030), the National Key Research and Development Project (No. 2018AAA0100300, No. 2022YFB3103104), and by China Postdoctoral Science Foundation (No. 2022M720320).

References

Cacheux, Y. L.; Borgne, H. L.; and Crucianu, M. 2019. Modeling Inter and Intra-Class Relations in the Triplet Loss for Zero-Shot Learning. In *Proceedings of the IEEE International Conference on Computer Vision*, 10333–10342.

Castro, F. M.; Marín-Jiménez, M. J.; Guil, N.; Schmid, C.; and Alahari, K. 2018. End-to-End Incremental Learning. In *Proceedings of European Conference on Computer Vision*, 233–248.

Chi, Z.; Gu, L.; Liu, H.; Wang, Y.; Yu, Y.; and Tang, J. 2022. MetaFSCIL: A Meta-Learning Approach for Few-Shot Class Incremental Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 14166–14175.

Chi, Z.; Mohammadi Nasiri, R.; Liu, Z.; Lu, J.; Tang, J.; and Plataniotis, K. N. 2020. All at Once: Temporally Adaptive Multi-Frame Interpolation with Advanced Motion Modeling. In *Proceedings of European Conference on Computer Vision*, 107–123.

Chi, Z.; Wang, Y.; Yu, Y.; and Tang, J. 2021. Test-Time Fast Adaptation for Dynamic Scene Deblurring via Meta-Auxiliary Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9137–9146.

Farhadi, A.; Endres, I.; Hoiem, D.; and Forsyth, D. 2009. Describing Objects by Their Attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1778–1785.

Feng, Y.; Huang, X.; Yang, P.; Yu, J.; and Sang, J. 2022. Non-generative Generalized Zero-shot Learning via Task-correlated Disentanglement and Controllable Samples Synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9346–9355.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of International Conference on Machine Learning*, 1126–1135.

Gautam, C.; Parameswaran, S.; Mishra, A.; and Sundaram, S. 2022. Tf-GCZSL: Task-Free Generalized Continual Zero-Shot Learning. *Neural Networks*, 155: 487–497.

Ghosh, S. 2021. Adversarial Training of Variational Auto-encoders for Continual Zero-shot Learning (A-CZSL). In *Proceedings of International Joint Conference on Neural Networks*, 1–8.

Gong, X.; Yuan, D.; and Bao, W. 2021. Discriminative Metric Learning for Partial Label Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 1–12.

Gong, X.; Yuan, D.; and Bao, W. 2022a. Partial Label Learning via Label Influence Function. In *Proceedings of International Conference on Machine Learning*, 7665–7678.

Gong, X.; Yuan, D.; and Bao, W. 2022b. Top-k Partial Label Machine. *IEEE Transactions on Neural Networks and Learning Systems*, 33(11): 6775–6788.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved Training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, 5767–5777.

Hou, S.; Pan, X.; Loy, C. C.; Wang, Z.; and Lin, D. 2019. Learning A Unified Classifier Incrementally via Rebalancing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 831–839.

Hu, X.; Tang, K.; Miao, C.; Hua, X.-S.; and Zhang, H. 2021. Distilling Causal Effect of Data in Class-Incremental Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3957–3966.

Kang, M.; Park, J.; and Han, B. 2022. Class-Incremental Learning by Knowledge Distillation with Adaptive Feature Consolidation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 16071–16080.

Koch, G.; Zemel, R.; Salakhutdinov, R.; et al. 2015. Siamese Neural Networks for One-Shot Image Recognition. In *Proceedings of International Conference on Machine Learning*.

Kong, X.; Gao, Z.; Li, X.; Hong, M.; Liu, J.; Wang, C.; Xie, Y.; and Qu, Y. 2022. En-Compactness: Self-Distillation Embedding & Contrastive Generation for Generalized Zero-Shot Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9306–9315.

Kuchibhotla, H. C.; Malagi, S. S.; Chandhok, S.; and Balasubramanian, V. N. 2022. Unseen Classes at a Later Time? No Problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9245–9254.

Lampert, C. H.; Nickisch, H.; and Harmeling, S. 2013. Attribute-based Classification for Zero-Shot Visual Object Categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3): 453–465.

Liu, W.; Xu, D.; Tsang, I. W.; and Zhang, W. 2018. Metric Learning for Multi-Output Tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2): 408–422.

Liu, Y.; Zhou, L.; Bai, X.; Huang, Y.; Gu, L.; Zhou, J.; and Harada, T. 2021. Goal-Oriented Gaze Estimation for Zero-Shot Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3794–3803.

Lyu, G.; Deng, X.; Wu, Y.; and Feng, S. 2022. Beyond Shared Subspace: A View-Specific Fusion for Multi-View Multi-Label Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 7647–7654.

Patterson, G.; and Hays, J. 2012. SUN Attribute Database: Discovering, Annotating, and Recognizing Scene Attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2751–2758.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. ICARL: Incremental Classifier and Representation Learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2001–2010.

Santoro, A.; Bartunov, S.; Botvinick, M.; Wierstra, D.; and Lillicrap, T. 2016. Meta-Learning with Memory-Augmented Neural Networks. In *Proceedings of International Conference on Machine Learning*, 1842–1850.

Schonfeld, E.; Ebrahimi, S.; Sinha, S.; Darrell, T.; and Akata, Z. 2019. Generalized Zero-and Few-Shot Learning via Aligned Variational Autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8247–8255.

Skorokhodov, I.; and Elhoseiny, M. 2021. Class Normalization for (Continual)? Generalized Zero-Shot Learning. In *Proceedings of International Conference on Learning Representations*.

Verma, V. K.; Liang, K. J.; Mehta, N.; and Carin, L. 2021. Meta-Learned Attribute Self-Gating for Continual Generalized Zero-Shot Learning. *arXiv preprint arXiv:2102.11856*.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; and Wierstra, D. 2016. Matching Networks for One Shot Learning. In *Advances in Neural Information Processing Systems*, 3630–3638.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The Caltech-Ucsd Birds-200-2011 Dataset.

Wei, K.; Deng, C.; Yang, X.; and Tao, D. 2022. Incremental Zero-Shot Learning. *IEEE Transactions on Cybernetics*, 52(12): 13788–13799.

Wei, K.; Deng, C.; Yang, X.; et al. 2020. Lifelong Zero-Shot Learning. In *Proceedings of International Joint Conference on Artificial Intelligence*, 551–557.

Wu, Y.; Liu, H.; Feng, S.; Jin, Y.; Lyu, G.; and Wu, Z. 2021. GM-MLIC: Graph Matching based Multi-Label Image Classification. In *Proceedings of International Joint Conference on Artificial Intelligence*, 1179–1185.

Xian, Y.; Lampert, C. H.; Schiele, B.; and Akata, Z. 2018a. Zero-Shot Learning—A Comprehensive Evaluation of The Good, The Bad and The Ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9): 2251–2265.

Xian, Y.; Lorenz, T.; Schiele, B.; and Akata, Z. 2018b. Feature Generating Networks for Zero-Shot Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5542–5551.

Yan, S.; Xie, J.; and He, X. 2021. Der: Dynamically Expandable Representation for Class Incremental Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3014–3023.