Meta-Learned Attribute Self-Gating for Continual Generalized Zero-Shot Learning

Vinay Kumar Verma [1] Kevin J Liang [1] Nikhil Mehta [1] Lawrence Carin [1]

Abstract

Zero-shot learning (ZSL) has been shown to be a promising approach to generalizing a model to categories unseen during training by leveraging class attributes, but challenges still remain. Recently, methods using generative models to combat bias towards classes seen during training have pushed the state of the art of ZSL, but these generative models can be slow or computationally expensive to train. Additionally, while many previous ZSL methods assume a one-time adaptation to unseen classes, in reality, the world is always changing, necessitating a constant adjustment for deployed models. Models unprepared to handle a sequential stream of data are likely to experience catastrophic forgetting. We propose a meta-continual zero-shot learning (MCZSL) approach to address both these issues. In particular, by pairing self-gating of attributes and scaled class normalization with meta-learning based training, we are able to outperform state-of-the-art results while being able to train our models substantially faster ($> 100\times$) than expensive generative-based approaches. We demonstrate this by performing experiments on five standard ZSL datasets (CUB, aPY, AWA1, AWA2 and SUN) in both generalized zero-shot learning and generalized continual zero-shot learning settings.

1. Introduction

Deep learning has demonstrated the ability to learn powerful models (Krizhevsky et al., 2012; He et al., 2016) given a sufficiently large, labeled dataset of a pre-defined set of classes. However, such models often generalize poorly when asked to classify previously unseen classes that were not encountered during training. In an ever-evolving world, in which new concepts or applications are to be expected, this brittleness can be an undesirable characteristic. While one could collect a new dataset and retrain a model, the associated time and costs make this rather inefficient.

In recent years, zero-shot learning (ZSL) (Akata et al., 2013; Norouzi et al., 2013; Mishra et al., 2017; Verma et al., 2020; Skorokhodov et al., 2021) has been proposed as an alternative framework. Rather than having to collect more data and relearn the network upon encountering a previously unseen class, zero-shot approaches seek to leverage auxiliary information about these new classes, often in the form of class attributes. This side information allows for reasoning about the relations between classes, enabling adaptation of the model to recognize samples from one of the novel classes. In the more general setting, a ZSL model should be capable of classifying inputs from both seen and unseen classes; this more difficult setting (Xian et al., 2018a;b; Verma et al., 2018) is commonly referred to as generalized zero-shot learning (GZSL).

Some of the strongest results in GZSL (Felix et al., 2018; Verma et al., 2018; Schonfeld et al., 2019; Xian et al., 2019a) have come from approaches utilizing generative models. These approaches typically learn a generative mapping between the attributes and the data and then conditionally generate synthetic samples from the unseen class attributes. The model can then be learned in the usual supervised manner on the joint dataset containing the seen class and generated unseen class data, mitigating model bias toward the seen classes. While effective, training the requisite generative model, generating synthetic data, and training the model on this combined data can be an expensive process (Narayan et al., 2020; Li et al., 2019; Verma et al., 2020) and requires all attributes of the unseen classes to be known *a priori*, limiting the flexibility of such approaches.

If assuming a one-time adaptation from a pre-determined set of training classes, the cost of such a one-off process may be considered acceptable in some circumstances, but it may present challenges if it must be repeated. While ZSL methods commonly consider only one adaptation, in reality environments are often dynamic, and new class data may appear sequentially. For example, if it is important for a model to be able to classify a previously unseen class, it is natural that a future data collection effort may later make labeled data from these classes available (Skorokhodov et al.,

[1] Duke University, NC, USA. Correspondence to: Vinay Verma <vinaykumar.verma@duke.edu >.

ArXiv Version: Under Review

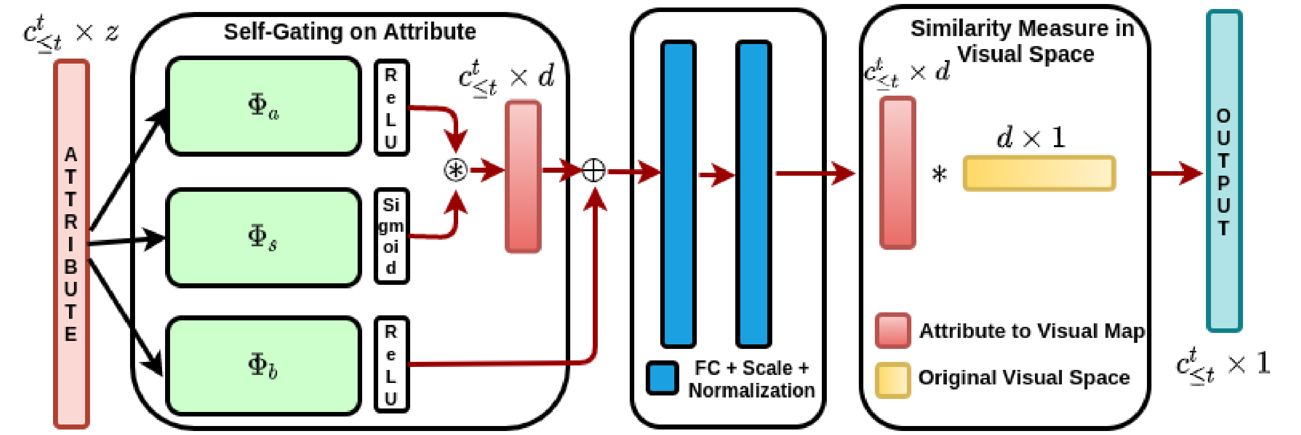

Figure 1. The proposed meta-continual zero-shot learning (MCZSL) model, with self-gating on attributes and scaled layer normalization.

2021). Alternatively, changing requirements may require the model to learn from and then generalize to entirely new seen and unseen classes (Gautam et al., 2020). In such cases, the model should be able to learn from new datasets without catastrophically forgetting (McCloskey & Cohen, 1989) previously seen data, even if that older data is no longer available in its entirety. Thus, it is important that ZSL methods can work in continual learning settings as well.

To address these issues, we propose meta-continual zero-shot learning (MCZSL). We propose a novel self-gating on the attributes and scaled layer normalization that obviate the need for expensive generative models, resulting in a $100\times$ speed-up in training time. We also show that training this model with a first-order meta-learning algorithm (Nichol et al., 2018) and a small reservoir of samples enables learning in a continual fashion, largely preventing catastrophic forgetting. Experiments on CUB (Wah et al., 2011), aPY (Farhadi et al., 2009), AWA1 (Lampert et al., 2009), AWA2 (Xian et al., 2018a) and SUN (Patterson & Hays, 2012) demonstrate that MCZSL achieves state-of-the-art results in both GZSL and continual GZSL settings.

2. Methods

The proposed approach can be divide into three major components: (i) self-gating on the attribute, which helps discard noisy attribute dimensions, providing canonical, robust, class-specific information; (ii) normalization and scale, which can mitigate the seen class bias and play a key role in zero-shot learning (Skorokhodov et al., 2021); and (iii) meta-learning for few-shot learning (Nichol et al., 2018), which enables the model to learn a robust mapping when only a few samples are present. We describe here notation and each component in detail.

2.1. Background and Notation

Let $T_t = \{\mathcal{D}^t_{tr}, \mathcal{D}^t_{ts}\}$ be a task arriving at time t, where $\mathcal{D}^t_{tr}$ and $\mathcal{D}^t_{ts}$ are the train and test sets associated with the t^{th} task, respectively. In a continual learning setup, we assume that a set of tasks arrive sequentially, such that the training data for only the current task is made available. Let this sequence of tasks be $\{T_1, T_2, \dots T_K\}$, where at time t, the training data for only the t^{th} task is available. In continual learning, the goal is to learn a new task while preventing catastrophic forgetting of any previous tasks, therefore, at test the model is evaluated on the current and previous tasks that the model has encountered before.

We assume that for a given task t, we have c^s_t number of seen classes that we can use for training and c^u_t unseen/novel classes that we are interested in adapting our model to. Analogous to GZSL, during test time we assume that samples come from any of the seen or unseen classes of the current or previous tasks, $i.e.$, the test samples for task T_t contain $c^s_{\le t}$ ($c^s_{\le t} = \sum_{k=1}^{t} c^s_k$) number of seen classes and c^u_t number of unseen classes, where c^u_t will depend on the chosen evaluation protocol (see Sections 4.1.2-4.1.3). Assume $\mathcal{D}^t_{tr} = \{x^t_i, y^t_i, a_{y_i}, t\}_{i=1}^{N^s_t}$ to be the training data, where $x^t_i \in \mathbb{R}^d$ is the visual feature of sample i of task t, y_i is the label of x^t_i, a_{y_i} is the attribute/description vector of label y_i, t is the task identifier (id) and $\mathcal{Y}_c$ is the label set for c classes. Let $\mathcal{D}^t_{ts} = \{\{x^t_i\}_{i=1}^{N^u_t}, y_{x^t_i}, \mathcal{A}_t : \mathcal{A}_t = \{a_y\}_{y=1}^{y=c^s_{\le t} \cup c^u_t}\}$ be the testing data for the task t and $y_{x^t_i} \in c^s_{\le t} \cup c^u_t$. For each task T_t, the seen and unseen classes are disjoint, $i.e.$, $\mathcal{Y}_{c^s_t} \cap \mathcal{Y}_{c^u_t} = \phi$. In each task t, seen and unseen classes have an associated attribute/description vector $a_y \in \mathcal{A}_t$ that helps to transfer knowledge from seen to unseen classes. We use N^s_t and N^u_t to denote the number of train and test samples for the task t, respectively. We assume that there are a total of S seen classes and U unseen classes for each dataset, and the attribute set is $\mathcal{A} : \mathcal{A} \in \mathbb{R}^{S \cup U \times z}$, where z is the attribute dimension. In the continual learning setup, test samples contain both unseen and seen classes up to task t. Therefore, the objective in continual ZSL is not only to predict the future task classes, but also to overcome catastrophic forgetting of the previous tasks.

2.2. Self-Gating on Attributes

Class attribute descriptions are a key component for positive forward transfer in ZSL. Models rely on the class description in order to successfully transfer knowledge from the seen classes to the unseen classes. These attributes are often either learned or manually annotated by humans, encapsulating all aspects of each class with a single set of attributes. Depending on the setting, intra-class variability can be significant; for example, members of a dog class can vary from a teacup poodle to Great Danes. As a result, setting global attributes that are representative of the entire class can be a significant challenge. Because of the diversity of samples within a class, certain dimensions of an attribute may contain noisy or redundant information, which can degrade model performance.

To address this, we develop a self-gating module for the attribute vector, that can continually learn from sequential data and provide a robust, class-representative vector satisfying most of the samples within the class. Let $\mathcal{A}^s_{\leq t}$ be the seen class attribute up to the training of task T_t, where $\mathcal{A}^s_{\leq t} \in \mathbb{R}^{c^s_{\leq t} \times D}$. We introduce Φ_a, Φ_s, and Φ_b as three neural networks with different activation functions at the output layer. The self-gating on the attribute can be defined as:

$$\begin{aligned}\mathcal{A}^s_{a_t^+} = \; & \text{ReLU}\left(\Phi_a\left(\mathcal{A}^s_{\leq t}\right)\right) * \sigma\left(\Phi_s\left(\mathcal{A}^s_{\leq t}\right)\right) \\ & + \text{ReLU}\left(\Phi_b\left(\mathcal{A}^s_{\leq t}\right)\right)\end{aligned} \tag{1}$$

where σ is the sigmoid activation function. The function Φ_s with a sigmoid activation maps the projected attribute to the range $[0, 1]$, giving weights to each dimension of the function Φ_a. Values closer to one signify higher importance of a particular attribute dimension, while values closer to zero imply the opposite. Finally, function Φ_b projects the attribute to the same space as Φ_a and Φ_b, resulting in something similar to a bias vector, but learned through a different function. We empirically observe that Φ_a and Φ_b help learn a robust and global attribute vector while Φ_b helps stabilize model training. An overview of the self-gating block is illustrated in Figure 1.

2.3. Scaled Layer Normalization

Normalization techniques have become critical components of deep learning models. Popular normalizations like BatchNorm (Ioffe & Szegedy, 2015), GroupNorm (Wu & He, 2018), and LayerNorm (Ba et al., 2016) not only help stabilize and accelerate training of deep learning models but can also lead to significant improvements in final performance. Recently, Skorokhodov et al. (2021) investigated the impact of class-wise layer normalization (CLN) on zero-shot learning. CLN was shown to result in significant improvement in ZSL models while also helping to overcome model bias towards the seen classes. The proposed scaled layer normalization (SLN) is motivated by CLN and scales the mean and variance vector by learnable parameters α and β. With $h_c \in \mathbb{R}^d$ being the output of the hidden layer of class c, we define SLN on h_c as:

$$SCN(h_c) = \frac{(h_c - \alpha\mu)}{\beta\sigma} \tag{2}$$

where $\{\alpha, \beta\} \in \mathbb{R}$ and $\{\mu, \sigma\} \in \mathbb{R}^d$. We apply SCN to each hidden layer, empirically observing that the proposed normalization leads to significant improvement.

2.4. Reservoir Sampling

In continual learning settings, we assume that data arrive sequentially task by task, with only samples of the present task available in their entirety; samples of classes from previous tasks ($T_1, \ldots, T_{t-1}$) are not accessible. Due to the tendency of neural networks to experience catastrophic forgetting (McCloskey & Cohen, 1989), the model is likely to forget previously learned knowledge while learning new tasks. While attribute self-gating and scaled layer normalization are highly effective for ZSL, by helping generalization to future classes, alone they do not prevent forgetting.

To mitigate catastrophic forgetting, our proposed model incorporates reservoir sampling (Vitter, 1985), using a small and constant memory size to store selected samples from previous tasks and replaying these samples while training the current task. This has several advantages: (i) each task can be trained with constant time and computational resources, and (ii) the number of samples do not grow as the tasks increase. Replaying samples from the reservoir effectively mitigates forgetting. For a simple reservoir, each sample is selected for replay with probability M/N, where M is the memory budget and N is the number of samples selected so far. We also tried using a ring buffer (Lopez-Paz et al., 2017), but empirically found this to provide similar performance as the simple reservoir we employ.

2.5. Training with Meta-Learning

While reservoir sampling is effective for mitigating forgetting, it also has several drawbacks. As the task number grows, a constant memory budget means the number of samples for of each class diminishes, as the same amount of memory has to accommodate a large number of classes. Similarly, the current task will have a sizably larger number of samples for each class in the current task than the number of samples in the reservoir for classes of past tasks.

Therefore, training can be difficult, resulting in models that generalize unevenly across all tasks/classes, as the model's learning is biased towards high-frequency classes. This issue is similar to the problem of few-shot learning, where we have to learn a model using only a few training examples present in the reservoir from the previous tasks.

To handle the aforementioned problems, we propose a meta-learning framework to train the model architecture. Meta-learning strategies have shown promising results for few-shot learning (Finn et al., 2017; Nichol et al., 2018). Some of these approaches do this by training the model to learn a better initialization, from which it can quickly adapt to novel tasks with only a few samples; as such, we view this to naturally complement reservoir sampling. Our approach leverages Reptile (Nichol et al., 2018), a relatively simple meta-learning approach using first-order gradient information, without storing gradients in memory for the inner loop. Let f be our proposed model with parameters θ, and τ be a batch of data in a N-way and K-shot setting (here we refer to batches instead of tasks to avoid confusion with tasks in a continual learning sense). Suppose $\mathcal{L}_t$ is the loss over the batch τ, given as:

$$\mathcal{L}_\tau = \mathbb{E}_{(x,y,a)\sim\tau} l(f_\theta(a,x),y) \tag{3}$$

where l may be any suitable loss; in our experiments, we use cross-entropy loss. Assume that U_τ^z is an operator representing z number of gradient update steps of the model parameter θ, on the batch of data τ. After z gradient updates of the loss $\mathcal{L}_\tau$, the model's new gradient is defined as:

$$\widetilde{\theta} = U_\tau^z(\theta) \tag{4}$$

After the final gradient update, instead of updating the gradient in the direction of $(\widetilde{\theta}-\theta)$, Reptile considers $(\theta-\widetilde{\theta})$ itself as a gradient, resulting in a final update:

$$\theta \leftarrow \theta - \eta(\theta - \widetilde{\theta}) \tag{5}$$

which has been shown by Nichol et al. (2018) to approximate model-agnostic meta-learning (Finn et al., 2017).

3. Related Work

We propose a novel approach to ZSL in the challenging setting where data arrives in a sequential manner. We cover here works in both ZSL and continual learning, as well as the few works that discuss both.

3.1. Zero-Shot Learning

The ZSL literature is both vast and diverse, with approaches that can be roughly divided into two categories: (i) non-generative and (ii)) generative approaches. Initial work (Akata et al., 2013; 2015; Norouzi et al., 2013; Hwang & Sigal, 2014; Fu et al., 2015; Xian et al., 2016) mainly focused on non-generative models. The objective of non-generative models is to learn a function from the seen classes that can measure the similarity between the visual and semantic spaces. (Akata et al., 2013; Norouzi et al., 2013; Lampert et al., 2014; Xian et al., 2016) measure the linear compatibility between the visual and semantic spaces; modeling the complex relation, linear compatibility is not as prominent. Another set of works (Zhang & Saligrama, 2016; Romera & Torr, 2015; Kodirov et al., 2015) focus on modeling relations by using bilinear compatibility relations, showing improved performance in similarity measure. These approaches show promising results for the ZSL setup where only unseen classes are evaluated on during test time, but when also simultaneously evaluated on seen classes ($i.e.$, GZSL), they perform poorly. This is primarily due to the inability to handle model bias towards the seen class samples. Recent works (Skorokhodov et al., 2021; Liu et al., 2021) have shown promising results for the GZSL setup using class normalization and isometric propagation networks. Purushwalkam et al. (2019) proposed a mechanism to use class attributes to gate modules that further process visual features for GZSL, which is reminiscent but different from the self-gating of attributes that we propose.

Of late, generative approaches have been among the most popular for GZSL. Because of the rapid progress in generative modeling ($e.g.$, VAEs (Kingma & Welling, 2014) and GANs (Goodfellow et al., 2014; Arjovsky et al., 2017)) generative approaches have been able to synthesize increasingly high-quality and realistic samples. For example, (Verma et al., 2018; Xian et al., 2018b; Mishra et al., 2017; Felix et al., 2018; Schonfeld et al., 2019; Chou et al., 2021; Xian et al., 2019b; Keshari et al., 2020) have used conditional VAEs or GANs to generate samples for unseen classes conditioned on the class attribute. These synthesized samples can then be used for training alongside samples from the seen classes, transforming ZSL into traditional supervised learning. Given the ability to generate as many samples as needed, these approaches can easily handle the model bias towards seen classes, leading to promising results for both ZSL and GZSL (Verma et al., 2020; Xian et al., 2019a).

3.2. Continual Learning

Catastrophic forgetting (McCloskey & Cohen, 1989; Carpenter & Grossberg, 1987; Lopez-Paz et al., 2017; Kirkpatrick et al., 2017; Jung et al., 2016; Rebuffi et al., 2017; Liang et al., 2018; Rajasegaran et al., 2020; KJ & Nallure Balasubramanian, 2020) is a key problem for neural network learning from streams of data, with previous data no longer available. Continual learning methods seek to balance the goals of mitigating catastrophic forgetting of previous tasks, learning new tasks, and transferring knowledge from previous tasks forward to allow for quicker adaptation in the future. The continual learning literature is often broadly divided into three categories: (i) replay-based (Rebuffi et al., 2017; Rajasegaran et al., 2020; KJ & Nallure Balasubramanian, 2020; van de Ven & Tolias, 2018), which rely on re-training the model with a small memory bank of samples from previous tasks; (ii) regularization-based (Kirkpatrick et al., 2017; Yu et al., 2020a; Lopez-Paz et al., 2017; Li & Hoiem, 2017), which regularize the model parameters to minimize deviation of parameters important to previous tasks while learning novel tasks; and (iii) expansion-based models (Rajasegaran et al., 2019; Masana et al., 2020; Xu & Zhu, 2018; Mallya et al., 2018; Singh et al., 2020; Mehta et al., 2021), which increase model capacity dynamically with each new task, preserving parameters for previous tasks. While there are many evaluation protocols commonly used, class incremental learning (Rajasegaran et al., 2020; Rebuffi et al., 2017), which does not assume a task identity (ID) during inference, is considered more realistic and challenging than the task incremental learning (Singh et al., 2020; Yoon et al., 2020), for which the task is known during inference.

3.3. Zero-Shot Continual Learning

One of the desiderata of continual learning techniques is forward transfer of previous knowledge to future tasks, which may not be known ahead of time; similarly, GZSL approaches seek to adapt models to novel, unseen classes while still being able to classify the seen classes. As such, there are clear connections between the two problem settings. Some recent works (Lopez-Paz et al., 2017; Wei et al., 2020; Skorokhodov et al., 2021; Gautam et al., 2020) have drawn increasing attention towards continual zero-shot learning (CZSL). For example, (Wei et al., 2020) considers a task incremental learning setting, where task ID for each sample is provided during train and test, an easier and perhaps less realistic setting than class incremental learning. A-GEM (Chaudhry et al., 2019) proposed a regularization-based model to overcome catastrophic forgetting while maximizing the forward transfer. (Skorokhodov et al., 2021) proposed a simple class normalization as an efficient solution to ZSL and extended it to CZSL; they proposed the setting discussed in Section 4.1.2. Meanwhile, (Gautam et al., 2020) proposed a replay-based approach for CZSL, showing state-of-the-art results in a more realistic setting discussed in Section 4.1.3. Like (Gautam et al., 2020; Skorokhodov et al., 2021), we follow the class incremental learning scenario and evaluate our approach in both settings.

4. Experiments

4.1. Training and Evaluation Protocols

4.1.1. Generalized Zero-Shot Learning (GZSL)

The simplest case we consider is the generalized zero-shot learning (GZSL) setting (Xian et al., 2018a). In GZSL, classes are split into two groups: classes whose data are available during the model's training stage ("seen" classes), and classes whose data only appear during inference ("unseen" classes). For both types, attribute vectors describing each class are available to facilitate knowledge transfer. During test time, samples may come from either classes seen during training or new unseen classes. We report mean seen accuracy (mSA) and mean unseen accuracy (mUA), as well as the harmonic mean (mhM) of both as an overall metric; harmonic mean is considered preferable to simple arithmetic mean as an overall metric, as it prevents either term from dominating (Xian et al., 2018a).

Note that some GZSL approaches (notably, generative ones) assume that the list of unseen classes and their attribute vectors are available during the training stage, even if their data are not; this inherently restricts these models to these known unseen classes. Conversely, our approach only requires the attributes of the seen classes. Also, in contrast to the continual GZSL settings described below, all seen classes are assumed available simultaneously during training.

4.1.2. Fixed Continual GZSL

The setting proposed by Skorokhodov et al. (2021) divides all classes of the dataset into K subsets, each corresponding to a task. For task T_t, the first t of these subsets are considered the seen classes, while the rest are unseen; this results in the number of seen classes increasing with t while the number of unseen classes decreases. Over the span of $t = 1, ...K$, this simulates a scenario where we eventually "collect" labeled data for classes that were previously unseen. Note that in contrast to the typical GZSL setting, only data from the t^{th} subset are available; previous training data are assumed inaccessible. The goal is to learn from this newly "collected" data without experiencing catastrophic forgetting. As in GZSL, we report mSA, mUA, and mH, but at the end of $K-1$ tasks:

$$mSA_F = \frac{1}{K-1}\sum_{i=1}^{K-1} \mathrm{Acc}(\mathcal{D}_{ts}^i(c^s_{\leq i}), \mathcal{A}(c^s_{\leq i})) \quad (6)$$

$$mUA_F = \frac{1}{K-1}\sum_{i=1}^{K-1} \mathrm{Acc}(\mathcal{D}_{ts}^i(c^u_{i}), \mathcal{A}(c^u_{i})) \quad (7)$$

$$mhM_F = \frac{1}{K-1}\sum_{i=1}^{K-1} \mathcal{H}(\mathcal{D}_{ts}^i(c^s_{\leq i}), \mathcal{D}_{ts}^i(c^u_{i}), \mathcal{A}) \quad (8)$$

where Acc represents per class accuracy, $\mathcal{D}_{ts}^i(c^s_{\leq i})$ and $\mathcal{A}(c^s_{\leq t})$ are the seen class test data and attribute vectors respectively during the i^{th} task. Similarly $\mathcal{D}_{ts}^i(c^u_i)$ and $\mathcal{A}(c^u_i)$ represents the unseen class test data and attribute vectors during the i^{th} task. $\mathcal{H}$ is the harmonic mean of the accuracies obtained on $\mathcal{D}_{ts}^i(c^s_{\leq i})$ and $\mathcal{D}_{ts}^i(c^u_i)$. We calculate the metric up to task $K-1$, as there are no unseen classes for task K, resulting in standard supervised continual learning.

4.1.3. Dynamic Continual GZSL

While it's not unreasonable that previously unseen class may become seen in the future, the above fixed continual GZSL evaluation protocol assumes that all unseen classes and attributes are set from the beginning, which may be un-

Table 1. Mean seen accuracy (mSA), mean unseen accuracy (mUA), and their harmonic mean (mH) for GZSL.

	SUN			CUB			AWA1			AWA2			Average
	mSA	mUA	mH	mSA	mUA	mH	mSA	mUA	mH	mSA	mUA	mH	Training Time
CVC-ZSL (Li et al., 2019)	36.3	42.8	39.3	47.4	47.6	47.5	62.7	77.0	69.1	56.4	81.4	66.7	3 Hours
SGAL (Yu & Lee, 2019)	42.9	31.2	36.1	47.1	44.7	45.9	52.7	75.7	62.2	55.1	81.2	65.6	50 Min
SGMA (Zhu et al., 2019)	–	–	–	36.7	71.3	48.5	-	-	-	37.6	87.1	52.5	–
DASCN (Ni et al., 2019)	42.4	38.5	40.3	45.9	59.0	51.6	59.3	68.0	63.4	–	–	–	–
TF-VAEGAN (Narayan et al., 2020)	45.6	40.7	43.0	52.8	64.7	58.1	–	–	–	59.8	75.1	66.6	1.75 Hours
EPGN (Yu et al., 2020b)	–	–	–	52.0	61.1	56.2	62.1	83.4	71.2	52.6	83.5	64.6	–
DVBE (Min et al., 2020)	45.0	37.2	40.7	53.2	60.2	56.5	–	–	–	63.6	70.8	67.0	–
LsrGAN (Vyas et al., 2020)	44.8	37.7	40.9	48.1	59.1	53.0	–	–	–	54.6	74.6	63.0	1.25 Hours
F-VAEGAN-D2 (Xian et al., 2019b)	45.1	38.0	41.3	48.4	60.1	53.6	–	–	–	57.6	70.6	63.5	–
ZSML (Verma et al., 2020)	45.1	21.7	29.3	60.0	52.1	55.7	57.4	71.1	63.5	58.9	74.6	65.8	3 Hours
NM-ZSL (Skorokhodov et al., 2021)	44.7	41.6	43.1	49.9	50.7	50.3	63.1	73.4	67.8	60.2	77.1	67.6	1 Min
MCZSL (Ours)	40.3	46.9	**43.4**	57.2	66.4	**61.4**	78.9	64.6	**71.8**	77.9	67.1	**72.1**	**31 Second**

Table 2. Mean seen accuracy (mSA), mean unseen accuracy (mUA), and their harmonic mean (mH) for fixed continual GZSL.

	CUB			aPY			AWA1			AWA2			SUN		
	mSA	mUA	mH	mSA	mUA	mH	mSA	mUA	mH	mSA	mUA	mH	mSA	mUA	mH
Sequential (Lower Bound)	11.44	2.84	4.25	36.53	15.78	17.66	40.91	12.08	18.11	43.37	12.02	18.12	11.82	3.06	4.78
Seq-CVAE (Mishra et al., 2017)	24.66	8.57	12.18	51.57	11.38	18.33	59.27	18.24	27.14	61.42	19.34	28.67	16.88	11.40	13.38
Seq-CADA (Schonfeld et al., 2019)	40.82	14.37	21.14	45.25	10.59	16.42	51.57	18.02	27.59	52.30	20.30	30.38	25.94	16.22	20.10
AGEM-CZSL (Chaudhry et al., 2019)	–	–	13.20	–	–	–	–	–	–	–	–	–	–	–	10.50
CZSL-CV+res (Gautam et al., 2020)	44.89	13.45	20.15	64.88	15.24	23.90	78.56	23.65	35.51	80.97	25.75	38.34	23.99	14.10	17.63
CZSL-CA+res (Gautam et al., 2020)	43.96	32.77	36.06	57.69	20.83	28.84	62.64	38.41	45.38	62.80	39.23	46.22	27.11	21.72	22.92
NM-ZSL (Skorokhodov et al., 2021)	55.45	43.25	47.04	45.26	21.35	27.18	70.90	37.46	48.75	76.33	39.79	51.51	50.01	19.77	28.04
MCZSL (Ours)	58.34	48.35	**51.31**	50.74	22.37	**30.66**	63.69	46.71	**53.12**	68.01	48.38	**55.17**	55.40	22.91	**32.23**

realistic. An alternative framing of continual GZSL is one in which each class consists of its own disjoint set of seen and unseen classes, as proposed by Gautam et al. (2020). Such a formulation does not require all attributes to be known *a priori*, allowing the model to continue accommodating an unbounded number of classes. As such, in contrast to the fixed continual GZSL, the number of seen and unseen classes both increase with t. As with the other settings, we report mSA, mUA and mH:

$$mSA_D = \quad \tfrac{1}{K}\textstyle\sum_{i=1}^{K} \mathrm{Acc}(\mathcal{D}_{ts}^{i}(c_{\leq i}^{s}), \mathcal{A}(c_{\leq i}^{s})) \tag{9}$$

$$mUA_D = \quad \tfrac{1}{K}\textstyle\sum_{i=1}^{K} \mathrm{Acc}(\mathcal{D}_{ts}^{i}(c_{\leq i}^{u}), \mathcal{A}(c_{\leq i}^{u})) \tag{10}$$

$$mhM_D = \quad \tfrac{1}{K}\textstyle\sum_{i=1}^{K} \mathcal{H}(\mathcal{D}_{ts}^{i}(c_{\leq i}^{s}), \mathcal{D}_{ts}^{i}(c_{\leq i}^{u}), \mathcal{A}) \tag{11}$$

where Acc represents per class accuracy, $\mathcal{D}_{ts}^{i}(c_{\leq i}^{s})$ and $\mathcal{A}(c_{\leq t}^{s})$ are the seen class test data and attribute vectors during i^{th} task. Similarly $\mathcal{D}_{ts}^{i}(c_{\leq i}^{u})$ and $\mathcal{A}(c_{\leq i}^{u})$ represents the unseen class test data and attribute vector during the i^{th} task. Detailed splits of the seen and unseen class samples for each task are given in the supplementary material.

4.2. Datasets and Baselines

Datasets We conduct experiments on five widely used datasets for zero-shot learning. CUB-200 (Wah et al., 2011) is a fine-grain dataset containing 200 classes of birds, and AWA1 (Lampert et al., 2009) and AWA2 (Xian et al., 2018a) are datasets containing 50 classes of animal, each represented by a 85-dimensional attribute. aPY (Farhadi et al., 2009) is a diverse dataset containing 32 classes, each associated with a 64-dimensional attribute. SUN (Patterson & Hays, 2012) contains 717 classes, each with only 20 samples; fewer samples and a high number of classes make SUN especially challenging. In the SUN dataset each class is represented by a 102-dimensional attribute vector. More dataset descriptions, hyperparameters, and implementation details can be found in the supplementary material.

Baselines We compare the proposed approach against a variety of baselines. CZSL-CV+res and CZSL-CA+res (Gautam et al., 2020) are continual ZSL models

Table 3. Mean seen accuracy (mSA), mean unseen accuracy (mUA), and their harmonic mean (mH) for dynamic continual GZSL.

	CUB			aPY			AWA1			AWA2			SUN		
	mSA	mUA	mH	mSA	mUA	mH	mSA	mUA	mH	mSA	mUA	mH	mSA	mUA	mH
Sequential (Lower Bound)	15.82	8.35	9.53	52.26	25.21	30.56	48.01	31.97	35.84	49.56	26.56	31.81	16.17	7.42	9.41
Seq-CVAE (Mishra et al., 2017)	38.95	20.89	26.74	65.87	17.90	25.84	70.24	28.36	39.32	73.71	26.22	36.30	29.06	21.33	24.33
Seq-CADA (Schmidhuber, 1987)	55.55	26.96	35.62	61.17	21.13	26.37	78.12	35.93	47.06	79.89	36.64	47.99	42.21	23.47	29.60
CZSL-CV+res (Gautam et al., 2020)	63.16	27.50	37.84	78.15	28.10	40.21	85.01	37.49	51.60	88.36	33.24	47.89	37.50	24.01	29.15
CZSL-CA+res (Gautam et al., 2020)	68.18	42.44	50.68	66.30	36.59	**45.08**	81.86	61.39	69.92	82.19	55.98	65.95	47.18	30.30	34.88
NM-ZSL (Skorokhodov et al., 2021)	64.91	46.05	53.79	79.60	22.29	32.61	75.59	60.87	67.44	89.22	51.38	63.41	50.56	35.55	41.65
MCZSL (Ours)	62.41	67.63	**64.71**	69.91	34.36	43.03	81.81	65.47	**72.22**	86.09	64.62	**72.96**	52.73	41.78	**46.45**

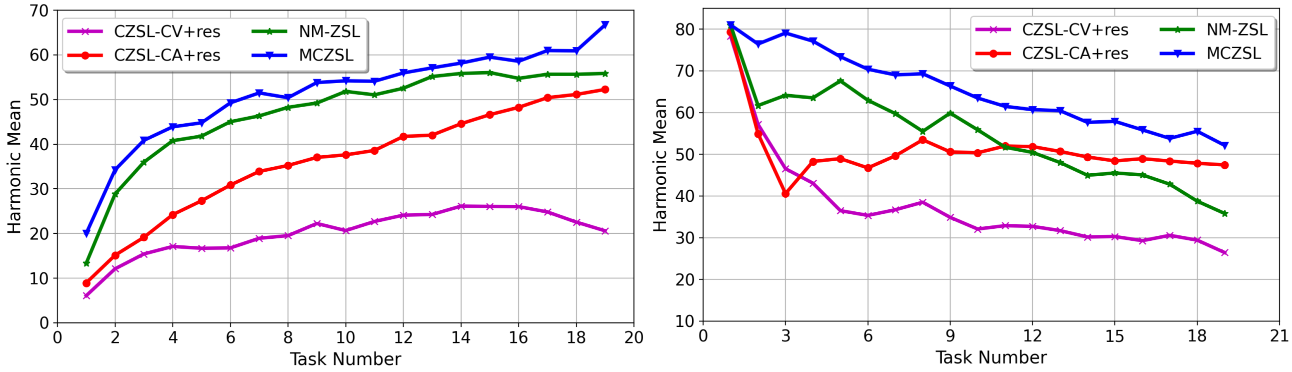

Figure 2. Harmonic mean per task for continual GZSL on the CUB dataset, Left: Fixed, Right: Dynamic

incorporating a conditional variational auto-encoders (VAE) and CADA (Schonfeld et al., 2019) with a memory reservoir, respectively. AGEM-CZSL (Chaudhry et al., 2019) is the average gradient episodic memory-based continual ZSL method. NM-ZSL (Skorokhodov et al., 2021) is an embedding-based approach that originally proposed what we refer to as fixed GZSL. Seq-CVAE and Seq-CADA are the sequential versions of VAE (Mishra et al., 2017) and CADA (Schonfeld et al., 2019) for CZSL settings, which can be considered as a lower bound of a generative approach.

4.3. Results

Generalized Zero-Shot Learning We conduct experiments on CUB-200, SUN, AWA1, and AWA2 datasets in the GZSL setting, reporting means of the seen and unseen classes, as well as their harmonic means, in Table 1. We observe significant improvements over previous methods across all datasets, with absolute gains of 0.3%, 2.3%, 0.6%, and 4.5% for SUN, CUB, AWA1, and AWA2 dataset. Particularly noteworthy is the average training[1] time of our approach relative to the baseline approaches, many of which require learning complex generative models ($e.g.$, VAE or GAN) to synthesize the realistic samples. Compared to other generative approaches, MCZSL can be 100-300× faster.

Fixed Continual GZSL We perform fixed continual GZSL experiments on the CUB, aPY, AWA1, AWA2 and SUN datasets, showing the results in Table 2. For fair comparison, we use the same memory buffer size as Gautam et al. (2020). On CUB, aPY, AWA1, AWA2 and SUN, the proposed model shows 4.27%, 1.82%, 4.37%, 3.66% and 4.19% absolute increase over the best baseline. Figure 2 (left) shows harmonic mean of the model versus the task number for the CUB dataset. With each task, more of the unseen classes become seen, so performance tends to increase over time. We observe that our proposed MCZSL consistently outperforms recent baselines. Notably, these significant improvements are achieved without using any costly generative models. Detailed descriptions of the experimental setup, task splits in each class, and memory reservoir size are given in the supplementary material.

Dynamic Continual GZSL As with fixed continual GZSL, we also conduct experiments in the dynamic continual GZSL on CUB, aPY, AWA1, AWA2 and SUN as shown in Table 3. Once again, we observe that our proposed MCZSL performs well, seeing absolute gains of 10.92%, 2.30%, 7.01%, and 4.80%, on the CUB, AWA1, AWA2, and

[1] Time to train on the whole dataset with a Nvidia GTX 1080Ti

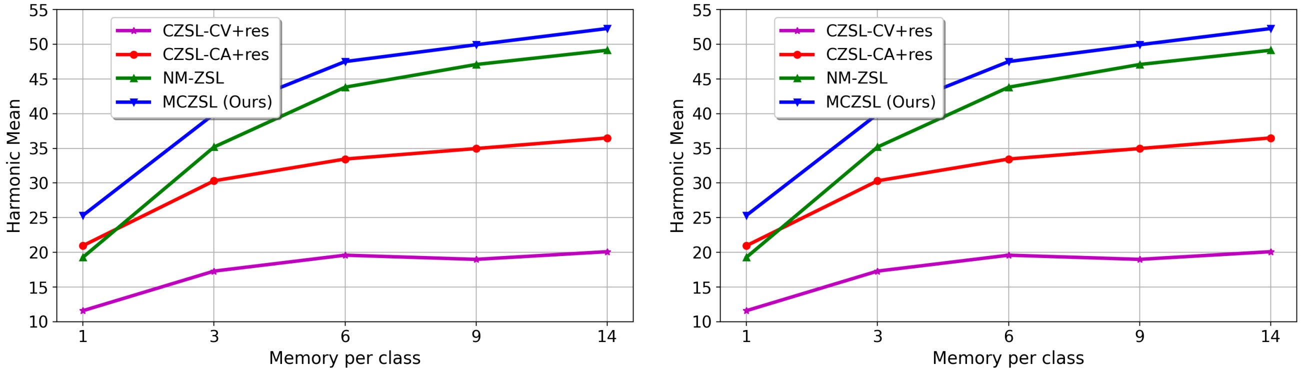

Figure 3. Memory growth vs model performance for continual GZSL on the CUB dataset, Left: Fixed, Right: Dynamic

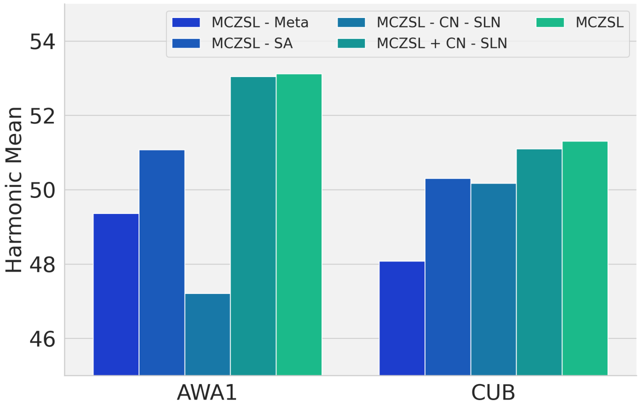

Figure 4. Effect of the different proposed components on the CUB and AWA1 datasets for fixed continual GZSL. *MCZSL*: our model with all components, *MCZSL-meta*: without meta-learning module, *MCZSL-SA*: without self-gating, *MCZSL-CN-SN*: without CN or SCN, and *MCZSL+CN-SCL*: with CN but without SCN.

SUN datasets, while being relatively competitive on aPY. In Figure 2 (right), we show harmonic mean per task for the CUB dataset. In contrast with fixed continual GZSL, each task brings more seen and unseen classes, making the problem harder due to more classes to distinguish and increasing opportunity for forgetting; thus, accuracy tends to drop with more tasks. Regardless, we again observe that the proposed model consistently outperforms recent baselines over the task numbers.

4.4. Ablation Studies

We conduct extensive ablation studies on the different components of the proposed model, observing that each of the proposed components play a critical role. We show the effects of different components on the AWA1 and CUB datasets in the fixed continual GZSL setting, with more ablation studies for dynamic continual GZSL in the supplementary material.

4.4.1. Reservoir size vs Performance

To overcome catastrophic forgetting, the model uses a constant-size reservoir (Lopez-Paz et al., 2017) to store previous task samples; with more tasks, the number of samples per class decreases. The reservoir size plays a key role for model performance. In Figure 3, we evaluate the model's performance for both fixed and dynamic continual GZSL. We observe that for different reservoir sizes $\{1, 3, 6, 9, 14\} \times \#classes$, the proposed model shows consistently better results compared to recent models. For fixed and dynamic continual GZSL, $\#classes$ is $S + U$ and S respectively.

4.4.2. Effect of Self-Gating on the Attribute

We apply self-gating to the learned embedding space of the attribute vector to get a robust and global representation. Ablations in Figure 4 show the effect of self-gating. We observe that our proposed MCZSL achieves harmonic means of 51.31 and 53.12 for the CUB and AWA1 datasets, respectively, which drop to 50.29 and 51.08 without self-gating.

4.4.3. Effect of Scaled Class Normalization

We observe that class normalization also plays a significant role to improve the model's performance. Removing the class normalization drops the harmonic mean from 53.12 to 47.21 and 51.31 to 50.17 for the AWA1 and CUB datasets respectively. Scaling the μ and σ using a learnable parameter α and β (as in (2)) slightly improves the model's performance. Refer to the Figure 4 for more details.

4.4.4. Effect of Meta-training

Because of its learning ability using the few samples and quick adaption, meta-learning (Nichol et al., 2018; Finn et al., 2017) based training plays a crucial role to improve the model performance. The meta-learning framework are described in the Section 2.5. We evaluate the model performance with and without meta-learning based training for fixed continual learning. The result are shown in the Figure 4. We empirically observe that if we withdraw the meta learning (MCZSL - meta) based training, the MCZSL performance drops significantly. On the AWA1 and CUB datasets in the fixed continual GZSL setting, the harmonic mean of MCZSL drops from 53.12 to 49.36 and 51.31 to 48.08 respectively.

5. Conclusions

We have proposed meta-continual zero-shot learning (MCZSL), a zero-shot learning method capable of operating in both generalized and continual learning settings. Through a novel self-gating on attributes and a scaled layer normalization, we obtain state-of-the art results in GZSL settings, despite not using expensive generative models; this allows for considerably faster speed during training, as well as flexibility to generalize to unlimited unseen classes that were unknown during training time. Given clear connections between zero-shot and continual learning, we also extend our approach to settings where data arrives sequentially. We adopt a data reservoir approach to mitigate catastrophic forgetting, and couple it with a meta-learning few-shot based approach to naturally enable the model to efficiently learn from the few samples that can be saved in a buffer. Our results on the CUB, aPY, AWA1, AWA2 and SUN datasets in GZSL and two different protocols of continual GZSL demonstrate that our approach outperforms a wide array of strong, recent baselines. Ablation studies demonstrate that each component of our proposed approach is critical for its success.

References

Akata, Z., Perronnin, F., Harchaoui, Z., and Schmid, C. Label-embedding for attribute-based classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 819–826, 2013.

Akata, Z., Reed, S., Walter, D., Lee, H., and Schiele, B. Evaluation of output embeddings for fine-grained image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2927–2936, 2015.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Carpenter, G. A. and Grossberg, S. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer vision, graphics, and image processing*, pp. 54–115, 1987.

Chaudhry, A., Ranzato, M., Rohrbach, M., and Elhoseiny, M. Efficient lifelong learning with a-gem. In *International Conference on Learning Representations*, 2019.

Chou, Y.-Y., Lin, H.-T., and Liu, T.-L. Adaptive and generative zero-shot learning. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=ahAUv8TI2Mz`.

Farhadi, A., Endres, I., Hoiem, D., and Forsyth, D. Describing objects by their attributes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1778–1785. IEEE, 2009.

Felix, R., Reid, I., Carneiro, G., et al. Multi-modal cycle-consistent generalized zero-shot learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 21–37, 2018.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135. PMLR, 2017.

Fu, Z., Xiang, T., Kodirov, E., and Gong, S. Zero-shot object recognition by semantic manifold distance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2635–2644, 2015.

Gautam, C., Parameswaran, S., Mishra, A., and Sundaram, S. Generalized continual zero-shot learning. *arXiv preprint arXiv:2011.08508*, 2020.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hwang, S. J. and Sigal, L. A unified semantic embedding: Relating taxonomies and attributes. *Advances in Neural Information Processing Systems*, 2014.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. PMLR, 2015.

Jung, H., Ju, J., Jung, M., and Kim, J. Less-forgetting learning in deep neural networks. *arXiv preprint arXiv:1607.00122*, 2016.

Keshari, R., Singh, R., and Vatsa, M. Generalized zero-shot learning via over-complete distribution. In *Proceedings* *of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13300–13308, 2020.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. Overcoming Catastrophic Forgetting in Neural Networks. *Proceedings of the National Academy of Sciences*, 2017.

KJ, J. and Nallure Balasubramanian, V. Meta-Consolidation for Continual Learning. *Advances in Neural Information Processing Systems*, 33, 2020.

Kodirov, E., Xiang, T., Fu, Z., and Gong, S. Unsupervised domain adaptation for zero-shot learning. In *The IEEE International Conference on Computer Vision*, pp. 2452–2460, 2015.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

Lampert, C. H., Nickisch, H., and Harmeling, S. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 951–958. IEEE, 2009.

Lampert, C. H., Nickisch, H., and Harmeling, S. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2014.

Li, K., Min, M. R., and Fu, Y. Rethinking zero-shot learning: A conditional visual classification perspective. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3583–3592, 2019.

Li, Z. and Hoiem, D. Learning without Forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

Liang, K. J., Li, C., Wang, G., and Carin, L. Generative Adversarial Network Training is a Continual Learning Problem. *arXiv preprint arXiv:1811.11083*, 2018.

Liu, L., Zhou, T., Long, G., Jiang, J., Dong, X., and Zhang, C. Isometric propagation network for generalized zero-shot learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=-mWcQVLPSPy.

Lopez-Paz, D., Ranzato, and Marc'Aurelio. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pp. 6467–6476, 2017.

Mallya, A., Davis, D., and Lazebnik, S. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 67–82, 2018.

Masana, M., Tuytelaars, T., and van de Weijer, J. Ternary feature masks: continual learning without any forgetting. *arXiv preprint arXiv:2001.08714*, 2020.

McCloskey, M. and Cohen, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Mehta, N., Liang, K. J., Verma, V. K., and Carin, L. Continual Learning using a Bayesian Nonparametric Dictionary of Weight Factors. *Artificial Intelligence and Statistics*, 2021.

Min, S., Yao, H., Xie, H., Wang, C., Zha, Z.-J., and Zhang, Y. Domain-aware visual bias eliminating for generalized zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12664–12673, 2020.

Mishra, A., Reddy, M., Mittal, A., and Murthy, H. A. A generative model for zero shot learning using conditional variational autoencoders. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop*, 2017.

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

Ni, J., Zhang, S., and Xie, H. Dual adversarial semantics-consistent network for generalized zero-shot learning. In *Advances in Neural Information Processing Systems*, pp. 6146–6157, 2019.

Nichol, A., Achiam, J., and Schulman, J. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Norouzi, M., Mikolov, T., Bengio, S., Singer, Y., Shlens, J., Frome, A., Corrado, G. S., and Dean, J. Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*, 2013.

Patterson, G. and Hays, J. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2751–2758. IEEE, 2012.

Purushwalkam, S., Nickel, M., Gupta, A., and Ranzato, M. Task-driven modular networks for zero-shot compositional learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3593–3602, 2019.

Rajasegaran, J., Hayat, M., Khan, S. H., Khan, F. S., and Shao, L. Random Path Selection for Continual Learning. *Advances in Neural Information Processing Systems*, 2019.

Rajasegaran, J., Khan, S., Hayat, M., Khan, F. S., and Shah, M. itaml: An incremental task-agnostic meta-learning approach. *arXiv preprint arXiv:2003.11652*, 2020.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. iCaRL: Incremental Classifier and Representation Learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

Romera, Paredes, B. and Torr, P. H. An embarrassingly simple approach to zero-shot learning. In *International Conference on Machine Learning*, pp. 2152–2161, 2015.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *IJCV*, pp. 211–252, 2015.

Schmidhuber, J. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.

Schonfeld, E., Ebrahimi, S., Sinha, S., Darrell, T., and Akata, Z. Generalized zero-and few-shot learning via aligned variational autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8247–8255, 2019.

Singh, P., Verma, V. K., Mazumder, P., Carin, L., and Rai, P. Calibrating cnns for lifelong learning. *Advances in Neural Information Processing Systems*, 33, 2020.

Skorokhodov, I., Mohamed, and Elhoseiny. Class normalization for zero-shot learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=7pgFL2Dkyyy.

van de Ven, G. M. and Tolias, A. S. Generative replay with feedback connections as a general strategy for continual learning. *arXiv preprint arXiv:1809.10635*, 2018.

Verma, V. K., Arora, G., Mishra, A., and Rai, P. Generalized zero-shot learning via synthesized examples. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

Verma, V. K., Brahma, D., and Rai, P. A meta-learning framework for generalized zero-shot learning. *Association for the Advancement of Artificial Intelligence*, 2020.

Vitter, J. S. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, pp. 37–57, 1985.

Vyas, M. R., Venkateswara, H., and Panchanathan, S. Leveraging seen and unseen semantic relationships for generative zero-shot learning. In *European Conference on Computer Vision*, pp. 70–86. Springer, 2020.

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

Wei, K., Deng, C., and Yang, X. Lifelong zero-shot learning. *IJCAI*, 2020.

Wu, Y. and He, K. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

Xian, Y., Akata, Z., Sharma, G., Nguyen, Q., Hein, M., and Schiele, B. Latent embeddings for zero-shot classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 69–77, 2016.

Xian, Y., Lampert, C. H., Schiele, B., and Akata, Z. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 2018a.

Xian, Y., Lorenz, T., Schiele, B., and Akata, Z. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018b.

Xian, Y., Sharma, S., Schiele, B., and Akata, Z. F-vaegan-d2: A feature generating framework for any-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019a.

Xian, Y., Sharma, S., Schiele, B., and Akata, Z. f-vaegan-d2: A feature generating framework for any-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10275–10284, 2019b.

Xu, J. and Zhu, Z. Reinforced continual learning. In *Advances in Neural Information Processing Systems*, pp. 899–908, 2018.

Yoon, J., Kim, S., Yang, E., and Hwang, S. J. Scalable and Order-robust Continual Learning with Additive Parameter Decomposition. *International Conference on Learning Representations*, abs/1902.09432, 2020.

Yu, H. and Lee, B. Zero-shot learning via simultaneous generating and learning. In *Advances in Neural Information Processing Systems*, pp. 46–56, 2019.

Yu, L., Twardowski, B., Liu, X., Herranz, L., Wang, K., Cheng, Y., Jui, S., and Weijer, J. v. d. Semantic drift compensation for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6982–6991, 2020a.

Yu, Y., Ji, Z., Han, J., and Zhang, Z. Episode-based prototype generating network for zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14035–14044, 2020b.

Zhang, Z. and Saligrama, V. Learning joint feature adaptation for zero-shot recognition. *arXiv preprint arXiv:1611.07593*, 2016.

Zhu, Y., Xie, J., Tang, Z., Peng, X., and Elgammal, A. Semantic-guided multi-attention localization for zero-shot learning. In *Advances in Neural Information Processing Systems*, pp. 14943–14953, 2019.

A. Dataset Descriptions

We conduct experiments on five widely used datasets for zero-shot learning. CUB-200 (Wah et al., 2011) is a fine-grain dataset containing 200 classes of birds, and AWA1 (Lampert et al., 2009) and AWA2 (Xian et al., 2018a) are datasets containing 50 classes of animal, each represented by an 85-dimensional attribute. aPY (Farhadi et al., 2009) is a diverse dataset containing 32 classes, each associated with a 64-dimensional attribute. SUN (Patterson & Hays, 2012) includes 717 classes, each with only 20 samples; fewer samples and a high number of classes make SUN especially challenging. In the SUN dataset, each class is represented by a 102-dimensional attribute vector. The details of the train/test split are summarized in Table-4; the same splits are used for the generalized zero-shot Learning (GZSL) setting.

We use the publicly available[2] pre-processed dataset provided by Xian et al. (2018a). For each visual domain, we use ResNet-101 (He et al., 2016) features pre-trained on ImageNet (Russakovsky et al., 2015). Features are directly extracted from the pre-trained model without any finetuning. Note that the seen and unseen splits proposed by (Xian et al., 2018a) ensure that unseen classes are not present in the ImageNet dataset; otherwise, the pre-trained weights would violates the zero-shot learning setting.

B. Reservoir Replay and Task Details for Fixed Continual GZSL

In the fixed continual GZSL setting, all classes appear during train and test for each class, either as seen or unseen. Initially, a small subset of classes are considered seen, while the rest are considered unseen; which each task, training data for an increasing number of the unseen classes becomes available, making them the new seen classes. For the AWA1 and AWA2 datasets, which contain $C^F_{\text{AWA}} = 50$ classes each, classes are divided into five tasks, with ten classes becoming seen per task. The memory reservoir we use is set to $25 \times C^F_{\text{AWA}}$. For SUN, we divide the $C^F_{\text{SUN}} = 717$ classes of SUN dataset into 15 tasks: 47 unseen classes becoming seen for the first three tasks, and 48 classes becoming seen for each of the remaining tasks. The memory reservoir for SUN is set to $5 \times C^F_{\text{SUN}}$. For the CUB dataset, which contains $C^F_{\text{CUB}} = 200$ classes, we divide all classes into 20 tasks, for ten classes becoming seen per task. We set memory reservoir to $10 \times C^F_{\text{CUB}}$. Finally, we split the $C^F_{\text{aPY}} = 32$ classes of the aPY dataset into four tasks of eight classes each. The memory reservoir for the aPY dataset is set to $25 \times^F_{\text{aPY}}$.

Dataset	Seen Classes	Unseen Classes	Attribute Dimension	Total Classes
AWA1	40	10	85	50
AWA2	40	10	85	50
CUB	150	50	312	200
SUN	645	72	102	717
aPY	20	12	64	32

Table 4. Datasets and their seen/unseen class splits for the dynamic GZSL setting.

C. Reservoir Replay and Task Details for Dynamic Continual GZSL

In the dynamic continual GZSL setting, the seen and unseen classes for each dataset are divided into multiple tasks, and with each task, the number of seen and unseen classes grows. Unlike the fixed continual GZSL setting, new seen classes are completely novel classes, not previously unseen ones. We split the AWA1 and AWA2 datasets to contain $C^D_{\text{AWA}} = 40$ seen classes and 10 unseen classes, evenly divided into five tasks; thus for each task, we have 8 seen and 2 unseen classes. The reservoir memory for the AWA1 and AWA2 datasets is set to $25 \times C^D_{\text{AWA}}$. The CUB dataset is split to have $C^D_{\text{CUB}} = 150$ seen and 50 unseen classes, divided into 20 tasks; the initial ten tasks are assigned seven seen and two unseen classes each, while the remaining ten tasks have eight seen and three unseen class. The reservoir memory for the CUB dataset is set to $10 \times C^D_{\text{CUB}}$. The aPY dataset has $C^D_{\text{aPY}} = 20$ seen and 12 unseen classes divided into four tasks, with each task having five seen classes and three unseen classes. The reservoir memory for the aPY dataset contains $25 \times C^D_{\text{aPY}}$. Similarly, the SUN dataset is split into 15 tasks: 43 seen and 4 unseen classes in the first three tasks, with the remaining 12 tasks containing 43 seen and five unseen classes. The reservoir memory for the SUN dataset contains $5 \times C^D_{\text{SUN}}$, where the number of seen classes is $C^D_{\text{SUN}} = 645$. We summarize the seen and unseen class splits and attribute dimension for each dataset in Table 4.

D. Implementation Details

For our experiments, we implement self-gating modules Φ_a, Φ_s, and Φ_b each as a single fully connected layer with ReLU, Sigmoid, and ReLU activation functions. The hidden dimension of each neural network module is set to 2048, on top of which scaled class normalization is applied. The self-gating output of a given attribute is then sent to another one-layer neural network of dimension $2048 \rightarrow 2048$, and scaled class normalization is applied again. This output is then considered the projected visual features. In this visual space, we measure similarity by cosine distance.

For each task of all datasets, the model is trained for 200

[2] http://datasets.d2.mpi-inf.mpg.de/xian/xlsa17.zip

epochs. For the inner training loop we use an Adam optimizer (Kingma & Ba, 2014) with a constant learning rate 0.0001. For the meta update, we use an Adam optimizer with an initial learning rate of 0.001, decreasing with each epoch at a rate of $1 - e/(E-1)$, where e and E are the current epoch and total number of epochs, respectively. We use the same hyperparameter settings for all dataset, finding our model to be stable, with applicability to a diverse set of datasets.