Neural Networks 155 (2022) 487–497

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Tf-GCZSL: Task-free generalized continual zero-shot learning

Chandan Gautam [a,*,1], Sethupathy Parameswaran [b,1], Ashish Mishra [c], Suresh Sundaram [b]

[a] Institute for Infocomm Research (I2R), A*STAR, Singapore
[b] Indian Institute of Science, Bangalore, India
[c] Indian Institute of Technology Madras, India

ARTICLE INFO

Article history:
Available online 6 September 2022

Keywords:
Zero-shot learning
Continual learning
Experience replay
Continual zero-shot learning
VAE

ABSTRACT

Learning continually from a stream of training data or tasks with an ability to learn the unseen classes using a zero-shot learning framework is gaining attention in the literature. It is referred to as continual zero-shot learning (CZSL). Existing CZSL requires clear task-boundary information during training which is not practically feasible. This paper proposes a task-free generalized CZSL (Tf-GCZSL) method with short-term/long-term memory to overcome the requirement of task-boundary in training. A variational autoencoder (VAE) handles the fundamental ZSL tasks. The short-term and long-term memory help to overcome the condition of the task boundary in the CZSL framework. Further, the proposed Tf-GCZSL method combines the concept of experience replay with dark knowledge distillation and regularization to overcome the catastrophic forgetting issues in a continual learning framework. Finally, the Tf-GCZSL uses a fully connected classifier developed using the synthetic features generated at the latent space of the VAE. The performance of the proposed Tf-GCZSL is evaluated in the existing task-agnostic prediction setting and the proposed task-free setting for the generalized CZSL over the five ZSL benchmark datasets. The results clearly indicate that the proposed Tf-GCZSL improves the prediction at least by 12%, 1%, 3%, 4%, and 3% over existing state-of-the-art and baseline methods for CUB, aPY, AWA1, AWA2, and SUN datasets, respectively in both settings (task-agnostic prediction and task-free learning). The source code is available at https://github.com/Chandan-IITI/Tf-GCZSL.

1. Introduction

Recently, conventional supervised learning frameworks in deep learning architecture have shown remarkable performance on various tasks (e.g., classification/recognition), computer vision (He, Zhang, Ren, & Sun, 2015), and natural language processing (Krizhevsky, Sutskever, & Hinton, 2012). Despite the recent success, conventional learning frameworks cannot handle unseen classes during testing or overcome the catastrophic forgetting problem while continuously learning to acquire new knowledge from a stream of data. Recently, the first limitation has been addressed by the zero-shot learning (ZSL) framework, where we classified objects from classes that are not available at the training time (Chao, Changpinyo, Gong, & Sha, 2016; Li et al., 2019; Xie et al., 2019; Zhu, Xie, Liu, & Elgammal, 2019). The continual learning framework can handle the second limitation of the conventional learning framework (Chaudhry, Dokania, Ajanthan and Torr, 2018; Kirkpatrick et al., 2017; Shin, Lee, Kim, & Kim, 2017). However, traditional ZSL approaches have difficulty with sequential training, and continual learning approaches cannot handle unseen classes. Therefore, a more preferable and desirable approach is needed to tackle sequential training and unseen classes problems simultaneously. This paper aims to leverage the advantages of both zero-shot learning and continual learning in a single framework.

Zero-shot learning (ZSL) is an interesting framework that has attracted considerable attention in recent years due to its ability to learn unseen/novel class examples. Earlier approaches for zero-shot learning are based on the embedding function between visual and semantic space and are therefore biased towards the seen classes. The generative models synthesize visual features directly from semantic class descriptors to address bias towards seen class issues. Feature generative methods provide a shortcut to cast the zero-shot learning problem into a conventional classification problem (Sohn, Lee, & Yan, 2015; Verma, Brahma, & Rai, 2020; Verma & Rai, 2017; Xian, Lorenz, Schiele, & Akata, 2018; Xian, Sharma, Schiele, & Akata, 2019; Yu, Ji, Han, & Zhang, 2020).

The conventional ZSL framework trains the model on different classes (of the same dataset) under the assumption that data

* Corresponding author.
E-mail addresses: gautamc@i2r.a-star.edu.sg (C. Gautam), sethupathyp@iisc.ac.in (S. Parameswaran), mishra@cse.iitm.ac.in (A. Mishra), vssuresh@iisc.ac.in (S. Sundaram).

[1] Equal Contribution.

https://doi.org/10.1016/j.neunet.2022.08.034
0893-6080/

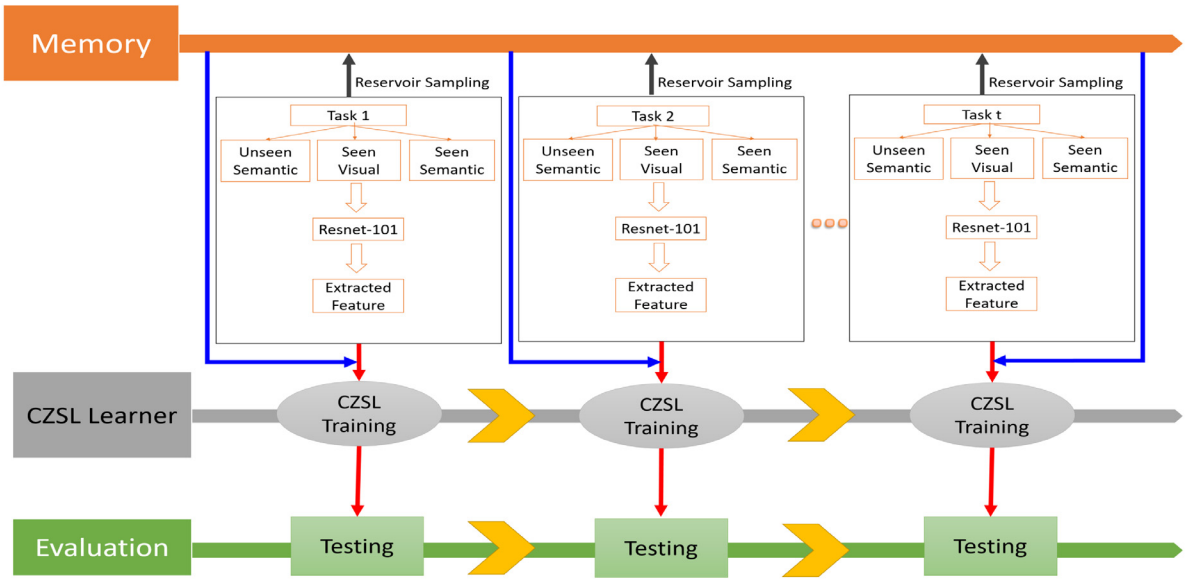


Fig. 1. A Generic CZSL framework.

for all the seen classes is available a priori. However, such an assumption is restrictive and not practical. In a practical setup, the training data arrives in a stream, and the training samples arrived in the stream may have samples from newly added (unseen) classes (see Fig. 1). One needs to update the ZSL model based on the newly arrived data. Otherwise, the prediction using the ZSL model will not be accurate. One way to overcome this issue is by retraining the ZSL model again, which is computationally intensive. Moreover, one needs to store the previous and current training data stream which requires a large memory.

Continual/incremental/lifelong learning (Chaudhry et al., 2019; Lopez-Paz & Ranzato, 2017; Shin et al., 2017) can address the above concern by enabling the sequential training of the ZSL model by preserving the accumulated (previous) knowledge while acquiring the new knowledge. This kind of method is known as continual zero-shot learning (CZSL). It can update its current knowledge continuously without forgetting previous information, in contrast to the conventional ZSL approaches. The CZSL method is a broad generalization of zero-shot learning. Here, it needs to be noted that in a traditional continual learning setting, training and testing data contain the same number of classes for classification. However, in the CZSL setting, training data also contains some unseen classes with their description in textual form, and a classifier should be able to classify these unseen classes during testing.

Most recently, a few CZSL methods (Skorokhodov & Elhoseiny, 2021; Wei, Deng, & Yang, 2020) have been proposed in the literature. Both of the existing methods in the literature (Skorokhodov & Elhoseiny, 2021; Wei et al., 2020) require task-boundary information during the training of the CZSL methods. The method proposed in Wei et al. (2020) considers one whole dataset as a task and trains a separate attribute encoder–decoder for each task (dataset); therefore, it is a very trivial setting (i.e., multi-head setting). Further, Skorokhodov and Elhoseiny (2021) develop an A-GEM-based CZSL method for a single-head setting; however, it is not a strict single-head setting as task identity is required during training. Nevertheless, in realistic situations, it is not always possible to get data with well-defined task boundaries (i.e., task identity). Further, there may be cases where we have access to only one sample at a time. These issues can be handled with task-free learning, which is closer to the realistic scenarios. Overall, both existing methods (Skorokhodov & Elhoseiny, 2021; Wei et al., 2020) do not support task-free learning setup of CZSL; therefore, the CZSL setup used in both papers is not suitable for a single-head setting. Hence, first time in the literature, this paper addresses a task free generalized CZSL (Tf-GCZSL) in a strict single-head setting where task identity is neither known during training nor during testing. For addressing this issue, Tf-GCZSL deploys two VAEs with knowledge distillation (KD), a long-term memory and a short-term memory. Here, KD and long-term memory help in alleviating the catastrophic forgetting and short-term memory makes the proposed method suitable for task-free learning.

The contributions of our proposed approach are summarized as:

1. To the best of our knowledge, this is the first work that proposes continual zero-shot learning for the task-free setting. The existing approaches (Skorokhodov & Elhoseiny, 2021; Wei et al., 2020) are only compatible when task-boundary is either present during training or during both training and testing.
2. To enable the model for task-free learning, this paper proposes two different task-free learning strategies based on short-term memory, which are compatible with any ZSL method.
3. To enable the generative model for CZSL, the proposed approach employs experience replay with KD. Here, we do not use the student–teacher network strategy for KD. Instead of that, we store the required information in the memory of the corresponding sample to perform KD. The stored information is generally known as dark knowledge (Hinton, Vinyals, & Dean, 2014).
4. This paper also provides the novel evaluation setting for CZSL as the existing settings (Skorokhodov & Elhoseiny, 2021; Wei et al., 2020) are not suitable for task-free learning.
5. Extensive experimental results validate the effectiveness of the proposed task-free CZSL method.

2. Related work

As CZSL mainly relies on continual learning and ZSL, this section briefly discusses both topics in two subsequent subsections.

2.1. Zero-shot learning

Recently, ZSL has attracted considerable attention due to handling unseen classes during testing. It transfers knowledge from seen classes to unseen classes via class attributes. Earlier proposed approaches for ZSL primarily were discriminative or non-generative (i.e., embedding-based) in nature (Akata, Perronnin, Harchaoui, & Schmid, 2016; Akata, Reed, Walter, Lee, & Schiele, 2015; Fu, Hospedales, Xiang, Fu, & Gong, 2014; Hwang & Sigal, 2014; Lampert, Nickisch, & Harmeling, 2013; Norouzi et al., 2013; Romera-Paredes & Torr, 2015; Socher, Ganjoo, Manning, & Ng, 2013; Xian et al., 2016; Zhang & Saligrama, 2015; Zhang, Xiang, & Gong, 2017). Non-generative methods learn an embedding from visual space to semantic space or vice versa via a linear compatibility function (Akata et al., 2016; Lampert et al., 2013; Norouzi et al., 2013; Xian et al., 2016). In contrast, generative models synthesize the examples for seen and unseen classes and transform a ZSL problem into a typical supervised learning problem (Felix, Kumar, Reid, & Carneiro, 2018; Huang, Wang, Yu, & Wang, 2019; Li et al., 2019; Schonfeld, Ebrahimi, Sinha, Darrell, & Akata, 2019a, 2019b; Verma, Arora, Mishra, & Rai, 2018; Xian et al., 2018, 2019; Zhu et al., 2019), which can be trained by any supervised classifiers.

2.2. Continual learning

Continual learning learns from streaming data with two objectives: avoiding catastrophic forgetting (preserving experience while learning on new tasks) and avoiding intransigence (updating new knowledge and transferring previous knowledge). The whole work of continual learning can be broadly categorized into three parts: (i) regularization-based methods (Chaudhry, Dokania et al., 2018; Kirkpatrick et al., 2017; Rebuffi, Kolesnikov, Sperl, & Lampert, 2017), (ii) replay-based methods (Chaudhry, Ranzato, Rohrbach and Elhoseiny, 2018; Chaudhry et al., 2019; Hayes, Cahill, & Kanan, 2019; Lopez-Paz & Ranzato, 2017; Shin et al., 2017), and (iii) parameter-isolation-based methods (Aljundi, Chakravarty, & Tuytelaars, 2017; Mallya, Davis, & Lazebnik, 2018; Mallya & Lazebnik, 2018; Rosenfeld & Tsotsos, 2018). Most of the earlier continual learning works are focused on multi-head setting (Chaudhry, Dokania et al., 2018; Kirkpatrick et al., 2017; Rebuffi et al., 2017). In recent years, task-free learning for traditional classification problem has received a surge of interest among researchers (Aljundi, Kelchtermans and Tuytelaars, 2019; Aljundi, Lin, Goujaud and Bengio, 2019; Buzzega, Boschini, Porrello, Abati, & Calderara, 2020; Jin, Du, & Ren, 2020) as it is a more practical continual learning setting than a multi-head setting. Instead of traditional classification problem, this paper focuses on task-free learning for the GZSL problem.

2.3. Continual zero-shot learning

In a traditional continual learning setting, training and testing data contain the same number of classes for classification. However, in the CZSL setting, training data also contains some unseen classes with their description in textual form, and a classifier should be able to classify these unseen classes during testing. Most recently, CZSL (Skorokhodov & Elhoseiny, 2021; Wei et al., 2020) has drawn increasing interest. To the best of our knowledge, only a handful the number of work is available for this problem. Chaudhry, Ranzato et al. (2018) developed an average gradient episodic memory (A-GEM) -based CZSL method for a multi-head setting. A generative model-based CZSL (Wei et al., 2020) method is also developed for multi-head setting. Most recently, Skorokhodov and Elhoseiny (2021) develop an A-GEM-based CZSL method for a single-head setting; however, it is not a strict single-head setting as task identity is required during training. This paper develops a CZSL method for a strict single-head setting where task identity is neither known during training nor testing.

3. Problem formulation

Formally, CZSL is divided among T tasks ($t \in 1, \ldots, T$), where each tth task consists of training and testing data stream. Generally, the training stream $\mathcal{D}_{tr}^t$ for tth task contains only the information of seen classes, which consists of feature vector x_i^t, task identity ι_i^t (it provides task-boundary), class label y_i^t, and class attribute information a_i^t. Where i represents ith sample from the whole training samples n_{tr} of tth task. In addition, training stream also contains class attribute information for unseen classes, i.e., $\mathcal{U}_{\mathcal{C}} = \{(a_i)_{i=1}^{n_{uc}}\}$ where n_{uc} denotes number of unseen classes. This is the key information which enables model for performing CZSL. Similarly, testing stream $\mathcal{D}_{ts}^t$ consists of $\{(x_i^t, \iota_i^t, y_i^t)_{i=1}^{n_{ts}}\}$, where n_{ts} is total number of test samples for tth task. Here, testing class label is only used for evaluation purpose. In this paper, we address single-head setting for two possible situations: (i) **task-agnostic prediction**: when task boundary is only available during training but not during testing, i.e., $\mathcal{D}_{tr}^t = \{(x_i^t, \iota_i^t, y_i^t, a_i^t)_{i=1}^{n_{tr}}\}$ and $\mathcal{D}_{ts}^t = \{(x_i^t, y_i^t)_{i=1}^{n_{ts}}\}$; (ii) **task-free learning**: when task boundary is neither available during training nor testing, i.e., $\mathcal{D}_{tr}^t = \{(x_i, y_i, a_i)_{i=1}^{n_{tr}}\}$ and $\mathcal{D}_{ts}^t = \{(x_i, y_i)_{i=1}^{n_{ts}}\}$.

4. Task-free generalized continual zero-shot learning: Tf-GCZSL

In this section, a task-free continual learning method is proposed for the GZSL framework, i.e., task-free generalized continual zero-shot learning (Tf-GCZSL). Tf-GCZSL is developed based on the concept of VAE, experience replay (ER) with KD, regularization, and short-term memory. The VAE helps in the GZSL tasks by generating synthetic features at the latent space (i.e., output of the encoder) and the output space (i.e., output of the decoder). The proposed Tf-GCZSL method (as shown in Fig. 2) deploys two distinct VAEs to process semantic and visual features separately. These VAEs generate discriminant features at the latent space by minimizing various losses simultaneously. This latent space information is used for classification and is utilized as a dark knowledge for performing KD, which helps in alleviating the problem of catastrophic forgetting. Here, KD is performed by using the dark knowledge (Hinton et al., 2014) instead of using the teacher network (i.e., the teacher network is the immediate previous network in the case of continual learning). Dark knowledge is the soft labels of the training samples of the previous tasks (Hinton et al., 2014), which is stored in the long-term memory for performing experience replay. This long-term memory also helps in alleviating catastrophic forgetting and regularizes the model for better performance. Along with this long-term memory, a short-term memory is also used to develop the Tf-GCZSL method for performing task-free learning with the CZSL framework. All the above-mentioned components are discussed in this section further. First of all, we discuss four kinds of losses, which are mainly deployed with VAEs for performing GZSL tasks in Tf-GCZSL. These losses are as follows:

Kullback–Leibler (KL) divergence and reconstruction loss: It minimizes two standard VAE losses simultaneously for the feature and the attribute encoder–decoder network: KL divergence (Kullback & Leibler, 1951) loss ($\mathcal{L}_{KL}$) and reconstruction loss ($\mathcal{L}_{Re}$).

Distribution-alignment loss (DA): It minimizes the difference in distribution between the latent space information of the feature and the attribute encoder.

$$\mathcal{L}_{DA} = (\|\mu_{Af} - \mu_{Vf}\|_2^2 + \|(\Sigma_{Af})^{\frac{1}{2}} - (\Sigma_{Vf})^{\frac{1}{2}}\|_{\mathcal{F}}^2)^{\frac{1}{2}}, \quad (1)$$

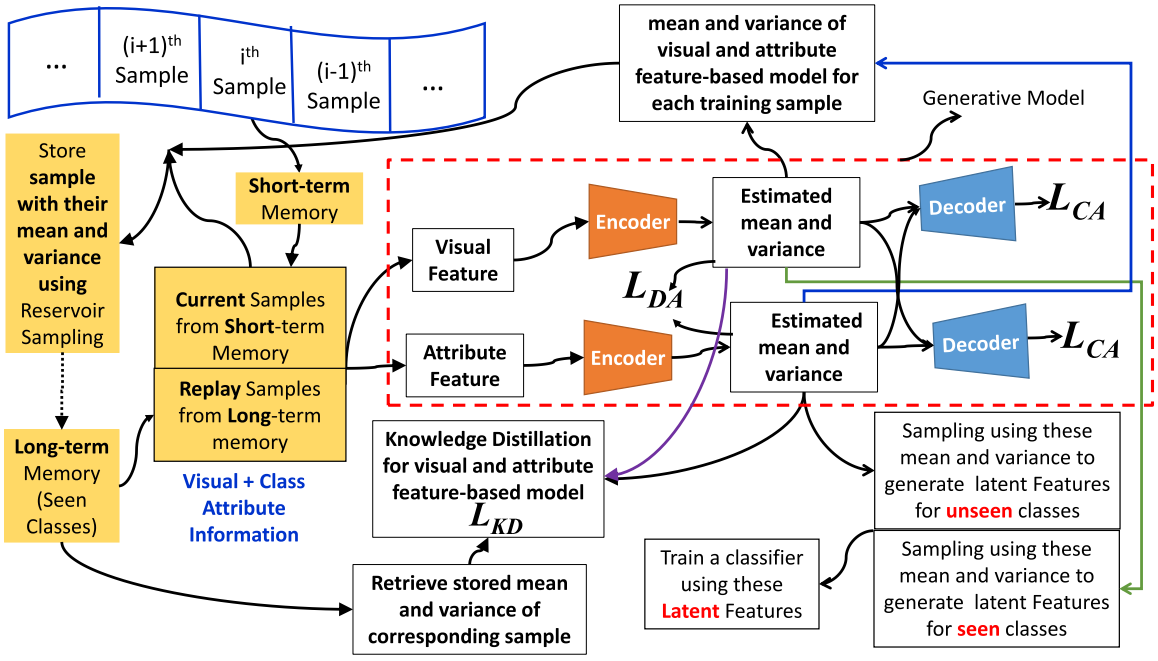


Fig. 2. Proposed Task-free-GCZSL framework. Here L_{DA} denotes the distribution alignment loss, L_{CA} denotes the cross alignment loss, and L_{KD} denotes the KD loss. Algorithms to use short-term memory is described in detail in the Algorithms 1 and 2.

where μ_{Vf} and Σ_{Vf} are the mean and variance estimated by visual encoder E_{Vf}, respectively. μ_{Af} and Σ_{Af} are the mean and variance estimated by the attribute encoder E_{Af}, respectively, and $\mathcal{F}$ represents Frobenius norm.

Cross-alignment loss (CA): It is cross-reconstruction loss between the output of the feature decoder and the attribute decoder and is given as

$$\mathcal{L}_{CA} = |a - D_{At}(E_{Vf}(x))| + |x - D_{Vf}(E_{Af}(a))|, \tag{2}$$

where x, a, D_{Af}, and D_{Vf} denote visual feature vector, class attribute vector, attribute decoder, and visual decoder, respectively.

The overall loss ($\mathcal{L}_G$) of a generative method for performing ZSL is as follows:

$$\mathcal{L}_G = \mathcal{L}_{Re} + \beta\mathcal{L}_{KL} + \gamma\mathcal{L}_{CA} + \delta\mathcal{L}_{DA}, \tag{3}$$

where β, γ and δ are the weighting factors.

4.1. ER and task-free strategies for CZSL

Experience Replay (ER) is a well-known method to alleviate catastrophic forgetting in the continual learning framework for handling the general classification task. However, in this paper, we combine ER and KD for task-free generalized continual zero-shot learning (Tf-GCZSL). In Tf-GCZSL, ER stores the previously learned samples in a small memory $\mathcal{M}$ and replays it later for training the model. The model is jointly trained by the samples from the replay memory $\mathcal{M}$ and the samples from the current streaming data. This joint training helps the model in retaining the past knowledge. Here, we need to address two important issues: (i) when the replay memory capacity $\mathcal{M}$ is full and (ii) task-free setting during training. In order to handle these issues, we employ reservoir sampling (Chaudhry et al., 2019; Vitter, 1985) which is a task-independent sampling technique. When the memory is full, reservoir sampling replaces an existing random sample in the memory with a new sample from the data stream with probability $\frac{\mathcal{M}}{l}$, where l is the number of samples seen so far. Further, the CZSL model needs to train in the task-free setting. In this setting, the samples arrive one by one to the model for training. However, training the model each time using a single sample can heavily overfit the CZSL model. Moreover, as the task boundary is unknown, it is difficult to optimize the model parameters and determine the stopping criteria. To handle these issues, we propose two different task-free learning strategies using short-term memory (it is to be noted that the short-term memory is different from the memory ($\mathcal{M}$) present in ER) as follows:

(i) Task-free CZSL strategy-1 (see Fig. 3): When the memory $\mathcal{M}$ reaches the maximum capacity for the first time, we stop the incoming data stream for a while and optimize the model once on the samples stored in the memory. After completing this one-time optimization, the training data stream resumes with a very small-sized short-term batch memory ($\mathcal{M}_b$) to store the incoming data stream. This short-term memory is simply a very small batch passed only once to the model for training without multiple epochs. After completing the training using $\mathcal{M}_b$, the memory is cleared to store other samples from the incoming data stream. The process is repeated until all the samples from the stream of training data are presented to the model. Since this strategy does not require multiple epochs, it is fast in learning the samples. It is referred to as Tf-GCZSL$_{\mathcal{M}_b}$. The pseudocode of this procedure is provided in Algorithm 1.

(ii) Task-free CZSL strategy-2 (see Fig. 4): In this strategy, we employ a larger short-term memory, i.e., $\mathcal{M}_{st}$ is larger than $\mathcal{M}_b$. The incoming samples are stored in $\mathcal{M}_{st}$ until it becomes full. Once this memory becomes full, we stop the incoming training samples for a while and train the model for multiple epochs for better generalization. After completing the training using $\mathcal{M}_{st}$, the $\mathcal{M}_{st}$ is cleared to store other samples from the incoming data stream. The process is repeated until there are no samples from

Algorithm 1 Task-free Learning Strategy-1

Input: Data stream $\mathcal{D}_{tr}$, Short-term memory $\mathcal{M}_b$
Output: Trained model

optimization_done ← False
for i^{th} sample in $\mathcal{D}_{tr}$ **do**
  Store the incoming i^{th} sample in replay memory $\mathcal{M}$ using Reservoir sampling strategy
  if Replay memory $\mathcal{M}$ is full and *optimization_done* is False **then**
    Stop the incoming data stream
    Train the CZSL model on $\mathcal{M}$ for multiple epochs on the available data in replay memory $\mathcal{M}$.
    optimization_done ← True
  else
    if *optimization_done* is True **then**
      Store i^{th} sample in a short-term batch memory $\mathcal{M}_b$
      if short-term batch memory $\mathcal{M}_b$ is full **then**
        Train the CZSL model continuously on the incoming batch of samples available in $\mathcal{M}_b$ and samples taken from replay memory $\mathcal{M}$ without running any epochs.
        Clear the short-term batch memory $\mathcal{M}_b$
      else
        Keep model in sleep for very small duration of time

the data stream. Tf-GCZSL with this strategy is referred to as Tf-GCZSL$_{\mathcal{M}_{st}}$. The pseudo-code of this procedure is provided in Algorithm 2.

Algorithm 2 Task-free Learning Strategy-2

Input: Data stream $\mathcal{D}_{tr}$, Short-term memory $\mathcal{M}_{st}$
Output: Trained model

for i^{th} sample in $\mathcal{D}_{tr}$ **do**
  Store the incoming i^{th} sample in replay memory $\mathcal{M}$ using Reservoir sampling strategy
  Store the incoming i^{th} sample in the short-term memory $\mathcal{M}_{st}$
  if M_{st} is full **then**
    Train the CZSL model on the samples taken from replay memory $\mathcal{M}$ and short-term memory $\mathcal{M}_{st}$ for multiple epochs to optimize the parameters
    Clear the short-term memory $\mathcal{M}_{st}$
  else
    Keep model in sleep until $\mathcal{M}_{st}$ is not full

4.2. Knowledge distillation using dark knowledge for CZSL

In addition to ER, Tf-GCZSL also performs KD with dark knowledge for mitigating catastrophic forgetting of the model. For this purpose, in addition to storing the training sample in $\mathcal{M}$, class attribute information and latent space information (i.e., estimated μ_{Vf}, Σ_{Vf} μ_{Af}, and Σ_{Af} by the encoder) corresponding to the training samples are also stored. This latent space information is dark knowledge, which is used to perform KD ($\mathcal{L}_{KD}^{dark}$) as:

$$\mathcal{L}_{KD}^{dark} = \|\mu_{Af} - \mu_{Af_{\mathcal{M}}}\|_1 + \|\mu_{Vf} - \mu_{Vf_{\mathcal{M}}}\|_1 + \|\Sigma_{Af} - \Sigma_{Af_{\mathcal{M}}}\|_1 + \|\Sigma_{Vf} - \Sigma_{Vf_{\mathcal{M}}}\|_1, \quad (4)$$

where $\mu_{Af_{\mathcal{M}}}$, $\mu_{Vf_{\mathcal{M}}}$, $\Sigma_{Af_{\mathcal{M}}}$ and $\Sigma_{Vf_{\mathcal{M}}}$ are retrieved from the stored latent information for the corresponding sample in $\mathcal{M}$. These values were estimated by the encoder at any point in time in the past on the learning trajectory of the Tf-GCZSL. One should note that the approach does not store/use any previously trained network as a teacher for performing KD. Instead, the knowledge required to perform distillation is stored in the $\mathcal{M}$ with sample information.

Table 1
Standard split of ZSL datasets.

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

4.3. Overall training procedure of Tf-GCZSL:

Overall, Tf-GCZSL minimizes the following loss:

$$\mathcal{L}_G = \mathcal{L}_{Re} + \beta\mathcal{L}_{KL} + \gamma\mathcal{L}_{CA} + \delta\mathcal{L}_{DA} + \alpha\mathcal{L}_{KD}^{dark}, \quad (5)$$

where β, γ, δ, and α are the weighting factors. For the task-free CZSL, first, minimize the loss and follow one of the two above-discussed task-free training strategies, i.e., either Tf-GCZSL$_{\mathcal{M}_b}$ or Tf-GCZSL$_{\mathcal{M}_{st}}$.

After completion of training, latent features are generated by sampling based on the mean and variance estimated by the visual/attribute encoder. The visual encoder is used to generate latent features for the seen classes, and the attribute encoder is used for the unseen classes. Since these latent features are very discriminative, a simple linear classifier using Softmax is trained on these latent features. The proposed Tf-GCZSL method can also be used for the task-agnostic prediction where the task boundary is known at the training time but not at the testing time. In this case, Tf-GCZSL minimizes the same loss function without the task-free learning strategy.

5. Performance evaluation

CZSL methods have been evaluated over five benchmark ZSL datasets, namely Caltech-UCSD-Birds 200–2011 (CUB) (Wah, Branson, Welinder, Perona, & Belongie, 2011), Attribute Pascal and Yahoo (aPY) (Farhadi, Endres, Hoiem, & Forsyth, 2009), Animals with Attributes (AWA1 and AWA2) (Farhadi et al., 2009), and SUN (Patterson & Hays, 2012). The standard split of these ZSL datasets is provided in Table 1. Here, we split these datasets and prepare them for two kinds of CZSL settings, which are discussed in the next subsection.

5.1. Settings and evaluation metrics

In the literature, two kinds of CZSL settings exist (Skorokhodov & Elhoseiny, 2021; Wei et al., 2020). The setting proposed in Wei et al. (2020) is a multi-head setting as shown in Fig. 5. Here, each dataset is considered as a separate task. Moreover, a distinct classifier and a distinct attribute encoder–decoder are deployed for each tasks, which is not feasible in real-time.

Another setting is proposed in Skorokhodov and Elhoseiny (2021), which is the CZSL setting for task-agnostic prediction as task information is known during training but not known during testing. In this section, we first discuss task-agnostic prediction for the CZSL setting and its limitation, then discuss about the new CZSL setting, i.e., task-free learning. Both experimental settings (task-agnostic and task-free CZSL settings) are designed based on the assumption of seen and unseen classes for each task. The details of each CZSL setting are provided below:

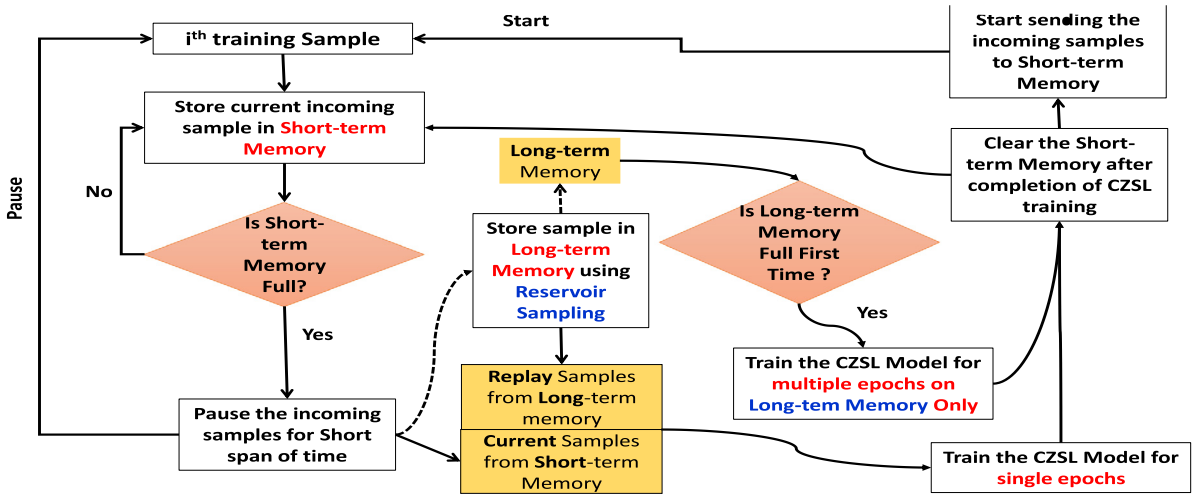


Fig. 3. Task free learning: Strategy-1.

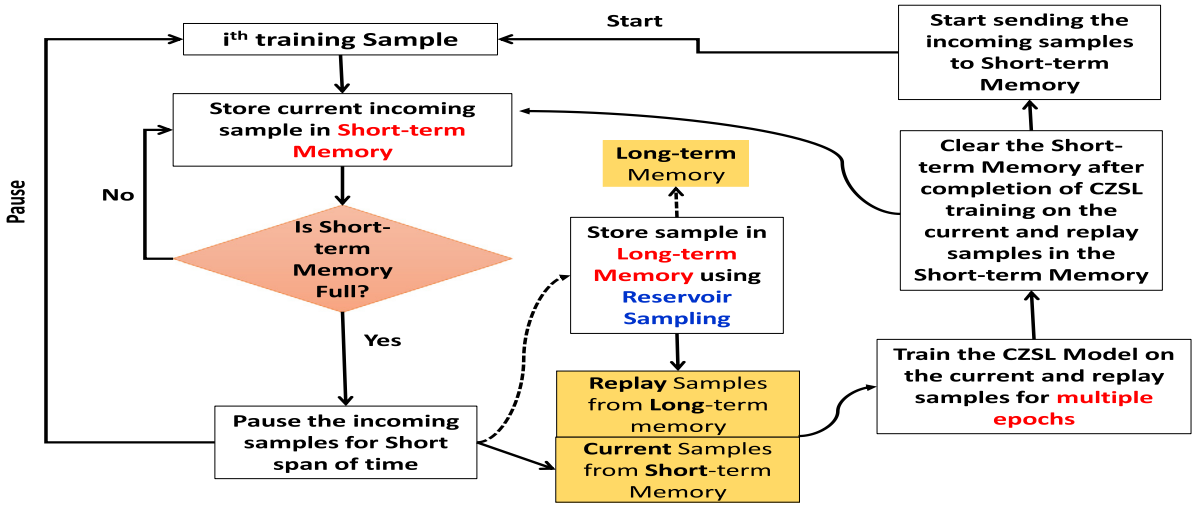


Fig. 4. Task free learning: Strategy-2.

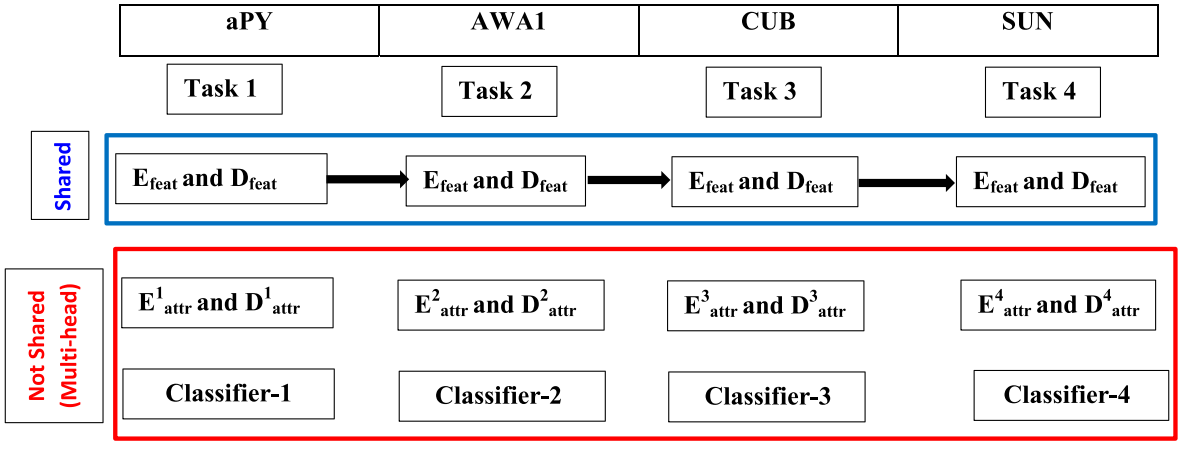
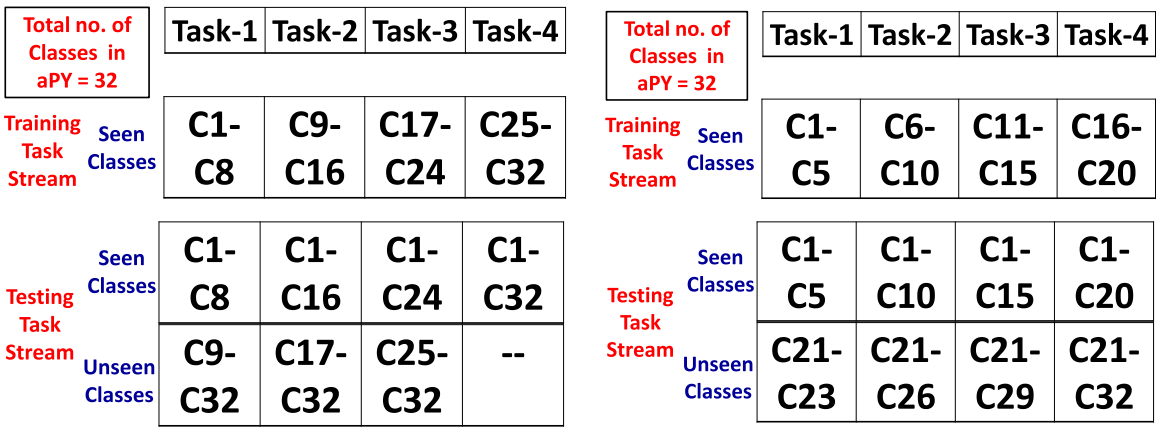


Fig. 5. CZSL setting in Wei et al. (2020).



(a) CZSL setting in [19]

(b) Proposed CZSL setting

Fig. 6. Task-agnostic prediction and task-free learning for CZSL settings.

CZSL setting for task-agnostic prediction:

For task-agnostic prediction, we have used the setting mentioned in Skorokhodov and Elhoseiny (2021) for CZSL. In this setting, first, data is divided among T tasks. Next, if the model is training on tth task, then all classes till tth task are treated as seen classes, and all classes from $(t+1)$th task to T tasks are treated as unseen classes. Following evaluation metrics are used to evaluate the model in case of task-agnostic prediction for the tth task (Skorokhodov & Elhoseiny, 2021):

- Mean Seen-class Accuracy (mSA)

$$mSA = \frac{1}{T}\sum_{t=1}^{T} CAcc(\mathcal{D}_{ts}^{\leq t}, A^{\leq t}), \tag{6}$$

where $CAcc$ stands for per class accuracy.

- Mean Unseen-class Accuracy(mUA)

$$mUA = \frac{1}{T-1}\sum_{t=1}^{T-1} CAcc(\mathcal{D}_{ts}^{>t}, A^{>t}) \tag{7}$$

- Mean Harmonic Accuracy (mH)

$$mH = \frac{1}{T-1}\sum_{t=1}^{T-1} H(\mathcal{D}_{ts}^{\leq t}, \mathcal{D}_{ts}^{>t}, A), \tag{8}$$

where H stands for harmonic mean.

Here, $\mathcal{D}^{\leq t}$ denotes all the train/test samples from 1st to tth task, and $D^{>t}$ denotes all the train/test samples from $(t+1)$th to last task.

Dataset division for task-agnostic prediction The 200 classes of CUB dataset are split into 20 tasks of 10 classes each. Similarly, the aPY dataset, which contains 32 classes, is split into 8 tasks with 4 classes each. The AWA1 and AWA2 datasets which have 50 classes each, is split into 10 tasks with 5 classes each. The SUN dataset has 717 classes and is difficult to split evenly. Hence, it is split into 15 tasks with 47 classes in the first 3 tasks and 48 classes in the remaining tasks. For all datasets, 20 percent of data from each task is taken as test data to compute the final evaluation metrics.

Limitation of the CZSL Setting in Skorokhodov and Elhoseiny (2021): Since all classes from all tasks are available as a seen or unseen class, the setting cannot be utilized for a class-incremental setup of continual learning. Note that it is an infeasible assumption that all classes' attribute information is known at the first task.

CZSL setting for task-free learning: The setting mentioned above in Skorokhodov and Elhoseiny (2021) is not suitable for task-free learning with CZSL, as seen and unseen classes are decided based on the task boundary (see Fig. 6(a)). However, in task-free learning, task boundary information is not available during the training and testing of the model. Therefore, we propose a more challenging and different CZSL setting for task-free learning, as shown in Fig. 6(b). In the figure, the task name is mentioned; however, we will not use it for task-free learning. We mentioned the task name so that the proposed setting can also be used to evaluate other CZSL methods which require task boundaries. Here, for task-free learning, data is split into multiple blocks based on the standard split of ZSL benchmark datasets and is explained in detail in the subsequent paragraph. Each block contains samples from distinct classes. First, we train the model by streaming samples from these blocks one by one, then test the model on the standard testing data available in the split of ZSL benchmark datasets. The performance is evaluated using the harmonic mean (H) and top-1 accuracy of seen-class accuracy (SA) and unseen-class accuracy (UA).

Dataset division for task-free learning In this setting, it needs to be noted that we divided the datasets into multiple blocks, but this block information is neither used during training nor testing because it is a task-free learning setting. These blocks are only for streaming the samples in a systematic manner. The CUB dataset is split into 20 blocks, with the first 10 blocks containing 7 seen classes and 3 unseen classes each and the next 10 blocks containing 8 seen classes and 2 unseen classes each. Here, the test data consists of the unseen classes and 20 percent data from seen classes of each block. The aPY dataset splits into 8 blocks, with the first 4 blocks containing 2 seen classes and 2 unseen classes each and the remaining blocks containing 3 seen classes and 1 unseen class each. The AWA1 and AWA2 datasets are split into 10 blocks with 4 seen classes and 1 unseen class per block. The SUN dataset splits into 15 blocks, with the first 3 blocks containing 43 seen classes and 4 unseen classes each and the remaining blocks containing 43 seen classes and 5 unseen classes each.

5.2. Baseline methods

There are only a handful of works available for CZSL. Recently, it is developed for multi-head setting (Wei et al., 2020) and task-agnostic prediction (Skorokhodov & Elhoseiny, 2021); however,

Table 2
CZSL results for task-agnostic prediction in terms of mean seen accuracy (mSA) for seen classes, mean unseen accuracy (mUA) for unseen classes, and their mean of harmonic mean (mH). The best results in the table are presented in boldface.

	CUB			aPY			AWA1			AWA2			SUN		
	mSA	mUA	mH	mSA	mUA	mH	mSA	mUA	mH	mSA	mUA	mH	mSA	mUA	mH
Seq-Tf-GCZSL	40.82	14.37	21.14	47.00	7.83	13.13	50.81	16.68	25.45	52.24	13.98	22.33	25.94	16.22	20.10
Seq-CVAE (Mishra, Krishna Reddy, Mittal, & Murthy, 2018)	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
AGEM+CZSL[a] (Chaudhry, Ranzato et al., 2018)	40.96	17.9	23.57	48.01	14.36	21.84	57.95	29.97	39.01	58.61	26.08	35.97	26.76	14.51	18.45
AGEM+CZSL+CN[a] (Skorokhodov & Elhoseiny, 2021)	36.98	18.34	23.71	37.33	22.86	28.21	62.07	34.55	42.74	61.52	35.73	43.73	27.62	17.99	21.25
EWC+CZSL[a] (Schwarz et al., 2018)	30.72	9.03	13.67	20.73	25.52	21.63	40.97	20.11	26.48	41.45	21.72	28.12	15.99	17.05	16.23
EWC+CZSL+CN[a] (Skorokhodov & Elhoseiny, 2021)	31.04	11.52	16.57	21.72	26.39	23.27	49.26	24.82	32.47	51.89	27.92	35.31	26.16	14.42	18.26
MAS+CZSL (Aljundi et al., 2018)	–	–	17.70	–	–	–	–	–	–	–	–	–	–	–	9.40
MAS+CZSL+CN (Skorokhodov & Elhoseiny, 2021)	–	–	23.80	–	–	–	–	–	–	–	–	–	–	–	14.20
GRCZSL (Gautam et al., 2021)	41.91	14.12	20.48	62.27	12.57	20.46	77.36	23.24	34.86	80.57	24.35	36.57	17.74	11.50	13.73
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
Tf-GCZSL$_{NK}$	45.00	30.50	34.57	58.41	18.74	26.85	61.67	37.38	44.90	65.46	36.40	45.75	27.07	23.35	23.84
Tf-GCZSL	46.63	32.42	**36.31**	57.92	21.22	**29.55**	64.00	38.34	**46.14**	64.89	40.23	**48.33**	28.09	24.70	**24.79**

[a]Indicates that the results are obtained by rerunning the respective methods in our proposed setting.

there is no work available for task-free learning. For task-agnostic prediction, the results are compared with the following methods:

- The sequential training of the proposed method without considering any continual learning setting: Seq-Tf-GCZSL.
- Skorokhodov et al. developed various methods for CZSL with and without class normalization (CN) (Skorokhodov & Elhoseiny, 2021):
  - (i) With CN: AGEM+CZSL+CN,[2] EWC+CZSL+CN, MAS+CZSL+CN
  - (ii) Without CN: AGEM+CZSL[1] (Chaudhry, Ranzato et al., 2018), EWC+CZSL (Schwarz et al., 2018), MAS+CZSL (Aljundi, Babiloni, Elhoseiny, Rohrbach, & Tuytelaars, 2018).
- GRCZSL (Gautam, Parameswaran, Mishra, & Sundaram, 2021): It is developed by using generative replay and a vanilla conditional variational autoencoder (CVAE).
- CZSL-CV+ res (Gautam, Parameswaran, Mishra, & Sundaram, 2020): It is developed by using a long-term memory-based experience replay strategy and CVAE.

For task-free learning, the results are compared with the offline training of the proposed method where one can assume all data are available at once, which is basically an upper bound for the proposed method. We also perform sequential training of the proposed methods: Seq-Tf-GCZSL$_{\mathcal{M}_b}$ and Seq-Tf-GCZSL$_{\mathcal{M}_{st}}$.

5.3. Results

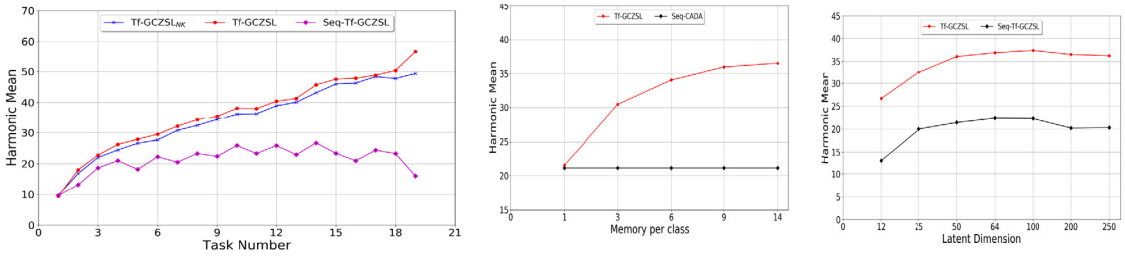
This section presents results for both cases of single head setting, i.e. task-agnostic prediction and task-free learning. The presented results for Tf-GCZSL are obtained using the hyperparameters and memory size mentioned in Tables 4 and 5.

Task-agnostic Prediction: Results for this setting are presented in Table 2 for 5 CZSL datasets. The performance results of Tf-GCZSL without KD are given in the table, i.e., Tf-GCZSL$_{NK}$. It can be observed from this table, Tf-GCZSL outperforms all existing CZSL methods presented in Skorokhodov and Elhoseiny (2021) by at least 12%, 1%, 3%, 4% and 3% margin for CUB, aPY, AWA1, AWA2 and SUN datasets in terms of *mH*, respectively. It also significantly outperforms the seq-Tf-GCZSL, which is obvious. Moreover, when we compare the results of Tf-GCZSL and Tf-GCZSL$_{NK}$, it has been observed that KD using dark knowledge with ER improves the performance and helps in alleviating the catastrophic forgetting further.

Task Free Learning: Results for this setting are presented in Table 3 for 5 CZSL datasets. We also provide the results of the proposed methods in the sequential setting (i.e., Seq-Tf-GCZSL$_{\mathcal{M}_b}$ and Seq-Tf-GCZSL$_{\mathcal{M}_{st}}$), and without 'KD using dark knowledge' (i.e., Tf-GCZSL$_{\mathcal{M}_b-NK}$ and Tf-GCZSL$_{\mathcal{M}_{st}-NK}$). It can be observed from this table that the proposed methods outperform sequential methods for all datasets. Further, dark knowledge improves the performance for most of the cases if we use $\mathcal{M}_{st}$. Overall, the second strategy-based task-free learning (Tf-GCZSL$_{\mathcal{M}_{st}}$) outperforms significantly over the first strategy (i.e., Tf-GCZSL$_{\mathcal{M}_b}$) by more than 5%, 6%, 2%, 9%, and 2% for CUB, aPY, AWA1, AWA2, and SUN datasets, respectively in terms of *H*. Moreover, Tf-GCZSL$_{\mathcal{M}_{st}}$ reaches closer to the upper bound (i.e., offline) as it lacks by only 8.55%, 9.28%, 4.38%, 1.49%, and 8.14% for CUB, aPY, AWA1, AWA2, and SUN datasets, respectively, in terms of *H*. This lack of performance is due to catastrophic forgetting.

The better performance of Tf-GCZSL in both settings is due to the joint training of the samples from the long-term (i.e., replay samples) and short-term (i.e., current samples) memories. Although the long-term memory does the repetitive training on the already trained samples, it does not lead to overfitting or

[2] It is to be noted that we performed class-balanced reservoir sampling for AGEM implementation.

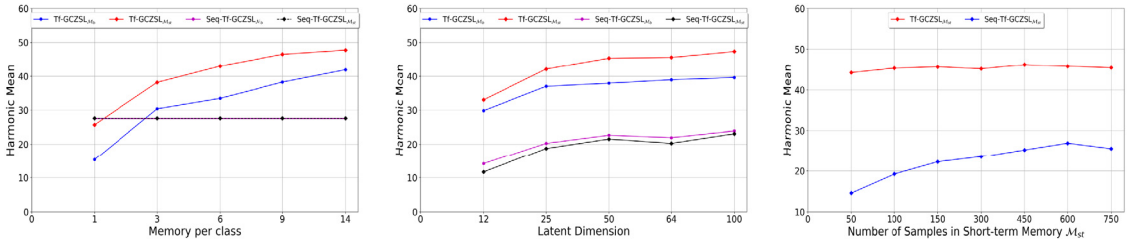


(a) Task-wise analysis

(b) Impact of the memory size $\mathcal{M}$ in terms of number of samples per class.

(c) Impact of the latent dimensions

Fig. 7. Ablation study for task-agnostic prediction.



(a) Impact of the size of the memory $\mathcal{M}$.

(b) Impact of the latent dimensions

(c) Impact of the size of $\mathcal{M}_{st}$

Fig. 8. Ablation study for task-free prediction.

Table 3
Results for task-free CZSL in terms of seen accuracy (SA) for seen classes, unseen accuracy (UA) for unseen classes, and their harmonic mean (H).

	CUB			aPY			AWA1			AWA2			SUN		
	SA	UA	H	SA	UA	H	SA	UA	H	SA	UA	H	SA	UA	H
Offline (Upper bound)	53.5	51.6	52.4	59.36	30.36	40.18	72.8	57.3	64.1	75	55.8	63.9	35.7	47.2	40.6
Seq-Tf-GCZSL$_{\mathcal{M}_b}$	38.73	21.42	27.58	23.51	1.68	3.14	20.37	2.12	3.85	14.64	9.43	11.47	27.89	18.88	22.52
Seq-Tf-GCZSL$_{\mathcal{M}_{st}}$	42.48	20.46	27.61	57.23	3.90	7.30	55.00	13.99	22.31	59.67	18.37	28.09	26.42	17.84	21.30
Tf-GCZSL$_{\mathcal{M}_b-NK}$	45.69	31.90	37.57	73.13	15.51	25.60	67.65	44.01	53.33	70.08	46.59	55.97	31.04	29.30	30.15
Tf-GCZSL$_{\mathcal{M}_{st}-NK}$	46.70	43.09	44.82	77.68	16.67	27.46	66.24	53.28	59.06	69.38	55.07	61.40	26.86	37.56	31.32
Tf-GCZSL$_{\mathcal{M}_b}$	45.08	34.02	38.78	72.55	14.33	23.94	65.64	51.46	57.69	68.42	42.74	52.62	31.00	29.37	30.16
Tf-GCZSL$_{\mathcal{M}_{st}}$	44.52	43.21	43.85	72.12	19.66	30.90	61.79	57.77	59.72	67.42	58.08	62.41	27.76	39.09	32.46

Table 4
Hyperparameters for Tf-GCZSL.

Parameters	aPY	AWA1	AWA2	CUB	SUN
Learning rate (VAE)	0.00015	0.00015	0.00015	0.00015	0.00015
Batch size (VAE)	50	50	50	50	50
Training epochs (VAE)	100	100	100	100	100
Learning rate (classifier)	0.001	0.001	0.001	0.001	0.001
Weight decay (classifier)	0.01	0.01	0.01	0.001	0.0003
Batch size (classifier)	32	32	32	32	32
Training epochs (classifier)	35	35	35	25	25

any adverse effect. In contrast, this joint training regularizes the model for better generalization.

5.4. Ablation study on CUB dataset

In this section, an ablation study is presented for the CUB dataset.

For task-agnostic prediction: The ablation study is presented in terms of the following three factors:

- **Task-wise analysis**: Task-wise analysis is depicted in Fig. 7(a). It can be observed from this figure that the performance of Tf-GCZSL improves as the number of tasks increases because as the number of tasks increases, then task-relatedness between seen and unseen class samples increases. Task-relatedness increases because the number of seen samples also increases, and it enriches the knowledge of the model. Although Tf-GCZSL improves performance when tasks increase, seq-Tf-GCZSL performance decreases as it does not use any continual learning strategy.
- **Analysis on replay memory**: The performance of Tf-GCZSL is very sensitive to the size of the memory. Size is kept in terms of the number of samples per class. If there are j number of classes and k number of samples per class, then the memory size is $j * k$. It can be observed from Fig. 7(b) that the performance of TF-GCZSL improves as memory size increases, as the memory can store more samples from the past experience.
- **Analysis on latent dimensions**: It is another important factor for CZSL. The performance of TF-GCZSL is depicted in 7(c) on different latent dimensions. This figure suggests that the size of latent dimensions should not be very small or very large. If it is very small, it cannot provide more discriminative features. If it is very large, then the degree of freedom increases, which will not provide compact features.

For task-free learning: Similarly, the ablation study is conducted on memory size and latent dimensions for task-free prediction. Since Tf-GCZSL uses two kinds of memories: replay memory $\mathcal{M}$ and short-term memory $\mathcal{M}_{st}$, analysis based on both memories is presented in Figs. 8(a) and 8(c) for $\mathcal{M}$ and $\mathcal{M}_{st}$, respectively. In the case of $\mathcal{M}$, performance increases as memory size increases

Table 5
Memory size used in Tf-GCZSL.

Parameters	aPY	AWA1	AWA2	CUB	SUN
$\mathcal{M}_{st}$	500	500	500	500	500
$\mathcal{M}_b$	50	50	50	50	50
$\mathcal{M}$	25 $*$ total classes	25 $*$ total classes	25 $*$ Total classes	10 $*$ total classes	5 $*$ total classes
Batch size from ER Memory	100	100	100	100	100

due to the same reason as discussed above. In the case of $\mathcal{M}_{st}$, size is not impacting much on the performance as these samples are jointly trained with the larger memory $\mathcal{M}$ in the task-free setting. Therefore, performance is very similar for all the cases. In the ablation of latent dimension in Fig. 8(b), we again observe the similar plot as 7(c). Moreover, for all three plots in Fig. 8, we also plot sequential results for better understanding, and the results are obvious that the proposed methods outperform sequential methods for all cases. Additionally, it is interesting to note that since samples are presented only once in Tf-GCZSL$_{\mathcal{M}_b}$ it has almost no overhead cost while Tf-GCZSL$_{\mathcal{M}_{st}}$ will add a little overhead cost.

6. Conclusion

This is the first work that tackles the continual Zero-shot learning for the task-free set-up to the best of our knowledge. This paper has proposed general task-free continual zero-shot learning strategies using VAE, ER using long-term memory, KD with dark knowledge, and two kinds of short-term memories. The performance is evaluated on five benchmark data, and the results indicate that the Tf-GCSZL achieves results that are closer to the upper bound with minimal catastrophic forgetting. The framework is generic; therefore, one can use other ZSL approaches to develop it for task-free CZSL.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We would like to thank the Wipro IISc Research and Innovation Network (WIRIN, India, Grant No-99325T), National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-RP-2021-027) for funding this research.

References

Akata, Zeynep, Perronnin, Florent, Harchaoui, Zaid, & Schmid, Cordelia (2016). Label-embedding for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(7), 1425–1438.

Akata, Zeynep, Reed, Scott, Walter, Daniel, Lee, Honglak, & Schiele, Bernt (2015). Evaluation of output embeddings for fine-grained image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. (pp. 2927–2936).

Aljundi, Rahaf, Babiloni, Francesca, Elhoseiny, Mohamed, Rohrbach, Marcus, & Tuytelaars, Tinne (2018). Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*. (pp. 139–154).

Aljundi, Rahaf, Chakravarty, Punarjay, & Tuytelaars, Tinne (2017). Expert gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. (pp. 3366–3375).

Aljundi, Rahaf, Kelchtermans, Klaas, & Tuytelaars, Tinne (2019). Task-free continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. (pp. 11254–11263).

Aljundi, Rahaf, Lin, Min, Goujaud, Baptiste, & Bengio, Yoshua (2019). Gradient based sample selection for online continual learning. In *Advances in neural information processing systems* (pp. 11816–11825).

Buzzega, Pietro, Boschini, Matteo, Porrello, Angelo, Abati, Davide, & Calderara, Simone (2020). Dark experience for general continual learning: a strong, simple baseline. *33*, In *Advances in neural information processing systems* (pp. 15920–15930).

Chao, Wei-Lun, Changpinyo, Soravit, Gong, Boqing, & Sha, Fei (2016). An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *ECCV* (pp. 52–68). Springer.

Chaudhry, Arslan, Dokania, Puneet K, Ajanthan, Thalaiyasingam, & Torr, Philip HS (2018). Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision (ECCV)*. (pp. 532–547).

Chaudhry, Arslan, Ranzato, Marc'Aurelio, Rohrbach, Marcus, & Elhoseiny, Mohamed (2018). Efficient lifelong learning with A-GEM. In *International conference on learning representations*.

Chaudhry, Arslan, Rohrbach, Marcus, Elhoseiny, Mohamed, Ajanthan, Thalaiyasingam, Dokania, Puneet K, Torr, Philip HS, et al. (2019). On tiny episodic memories in continual learning. arXiv preprint arXiv:1902.10486.

Farhadi, Ali, Endres, Ian, Hoiem, Derek, & Forsyth, David (2009). Describing objects by their attributes. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 1778–1785). IEEE.

Felix, Rafael, Kumar, Vijay BG, Reid, Ian, & Carneiro, Gustavo (2018). Multi-modal cycle-consistent generalized zero-shot learning. In *ECCV* (pp. 21–37).

Fu, Yanwei, Hospedales, Timothy M, Xiang, Tao, Fu, Zhenyong, & Gong, Shaogang (2014). Transductive multi-view embedding for zero-shot recognition and annotation. In *European conference on computer vision* (pp. 584–599). Springer.

Gautam, Chandan, Parameswaran, Sethupathy, Mishra, Ashish, & Sundaram, Suresh (2020). Generalized continual zero-shot learning. arXiv preprint arXiv:2011.08508.

Gautam, Chandan, Parameswaran, Sethupathy, Mishra, Ashish, & Sundaram, Suresh (2021). Generative replay-based continual zero-shot learning. arXiv preprint arXiv:2101.08894.

Hayes, Tyler L., Cahill, Nathan D., & Kanan, Christopher (2019). Memory efficient experience replay for streaming learning. In *2019 international conference on robotics and automation (ICRA)* (pp. 9769–9776). IEEE.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, & Sun, Jian (2015). Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385.

Hinton, Geoffrey, Vinyals, Oriol, & Dean, Jeff (2014). Dark knowledge. In *Presented as the keynote in baylearn, Vol. 2*.

Huang, He, Wang, Changhu, Yu, Philip S., & Wang, Chang-Dong (2019). Generative dual adversarial network for generalized zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. (pp. 801–810).

Hwang, Sung Ju, & Sigal, Leonid (2014). A unified semantic embedding: Relating taxonomies and attributes. In *Advances in neural information processing systems* (pp. 271–279).

Jin, Xisen, Du, Junyi, & Ren, Xiang (2020). Gradient based memory editing for task-free continual learning. In *International conference on machine learning workshops*.

Kirkpatrick, James, Pascanu, Razvan, Rabinowitz, Neil, Veness, Joel, Desjardins, Guillaume, Rusu, Andrei A, et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, *114*(13), 3521–3526.

Krizhevsky, Alex, Sutskever, Ilya, & Hinton, Geoffrey E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Kullback, Solomon, & Leibler, Richard A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, *22*(1), 79–86.

Lampert, Christoph H., Nickisch, Hannes, & Harmeling, Stefan (2013). Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*(3), 453–465.

Li, Jingjing, Jing, Mengmeng, Lu, Ke, Ding, Zhengming, Zhu, Lei, & Huang, Zi (2019). Leveraging the invariant side of generative zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. (pp. 7402–7411).

Lopez-Paz, David, & Ranzato, Marc'Aurelio (2017). Gradient episodic memory for continual learning. In *Advances in neural information processing systems* (pp. 6467–6476).

Mallya, Arun, Davis, Dillon, & Lazebnik, Svetlana (2018). 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).

Mallya, Arun, & Lazebnik, Svetlana (2018). Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. (pp. 7765–7773).

Mishra, Ashish, Krishna Reddy, Shiva, Mittal, Anurag, & Murthy, Hema A (2018). A generative model for zero shot learning using conditional variational autoencoders. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. (pp. 2188–2196).

Norouzi, Mohammad, Mikolov, Tomas, Bengio, Samy, Singer, Yoram, Shlens, Jonathon, Frome, Andrea, et al. (2013). Zero-shot learning by convex combination of semantic embeddings. arXiv preprint arXiv:1312.5650.

Patterson, Genevieve, & Hays, James (2012). Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *2012 IEEE conference on computer vision and pattern recognition* (pp. 2751–2758). IEEE.

Rebuffi, Sylvestre-Alvise, Kolesnikov, Alexander, Sperl, Georg, & Lampert, Christoph H (2017). icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. (pp. 2001–2010).

Romera-Paredes, Bernardino, & Torr, Philip (2015). An embarrassingly simple approach to zero-shot learning. In *International conference on machine learning* (pp. 2152–2161).

Rosenfeld, Amir, & Tsotsos, John K. (2018). Incremental learning through deep adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(3), 651–663.

Schonfeld, Edgar, Ebrahimi, Sayna, Sinha, Samarth, Darrell, Trevor, & Akata, Zeynep (2019). 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).

Schonfeld, Edgar, Ebrahimi, Sayna, Sinha, Samarth, Darrell, Trevor, & Akata, Zeynep (2019). 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).

Schwarz, Jonathan, Czarnecki, Wojciech, Luketina, Jelena, Grabska-Barwinska, Agnieszka, Teh, Yee Whye, Pascanu, Razvan, et al. (2018). Progress & compress: A scalable framework for continual learning. In *International conference on machine learning* (pp. 4528–4537). PMLR.

Shin, Hanul, Lee, Jung Kwon, Kim, Jaehong, & Kim, Jiwon (2017). Continual learning with deep generative replay. In *Advances in neural information processing systems* (pp. 2990–2999).

Skorokhodov, Ivan, & Elhoseiny, Mohamed (2021). Normalization matters in zero-shot learning. In *International conference on learning representations*.

Socher, Richard, Ganjoo, Milind, Manning, Christopher D, & Ng, Andrew (2013). Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems* (pp. 935–943).

Sohn, Kihyuk, Lee, Honglak, & Yan, Xinchen (2015). Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems* (pp. 3483–3491).

Verma, Vinay Kumar, Arora, Gundeep, Mishra, Ashish, & Rai, Piyush (2018). Generalized zero-shot learning via synthesized examples. In *CVPR*.

Verma, Vinay Kumar, Brahma, Dhanajit, & Rai, Piyush (2020). Meta-learning for generalized zero-shot learning. In *AAAI* (pp. 6062–6069).

Verma, Vinay Kumar, & Rai, Piyush (2017). A simple exponential family framework for zero-shot learning. In *ECML-PKDD* (pp. 792–808).

Vitter, Jeffrey S. (1985). Random sampling with a reservoir. *ACM Transactions on Mathematical Software*, *11*(1), 37–57.

Wah, Catherine, Branson, Steve, Welinder, Peter, Perona, Pietro, & Belongie, Serge (2011). The caltech-ucsd birds-200–2011 dataset.

Wei, Kun, Deng, Cheng, & Yang, Xu (2020). Lifelong zero-shot learning. In *Proceedings of the twenty-ninth international joint conference on artificial intelligence*. (pp. 551–557).

Xian, Yongqin, Akata, Zeynep, Sharma, Gaurav, Nguyen, Quynh, Hein, Matthias, & Schiele, Bernt (2016). Latent embeddings for zero-shot classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. (pp. 69–77).

Xian, Yongqin, Lorenz, Tobias, Schiele, Bernt, & Akata, Zeynep (2018). Feature generating networks for zero-shot learning. In *CVPR* (pp. 5542–5551).

Xian, Yongqin, Sharma, Saurabh, Schiele, Bernt, & Akata, Zeynep (2019). 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).

Xie, Guo-Sen, Liu, Li, Jin, Xiaobo, Zhu, Fan, Zhang, Zheng, Qin, Jie, et al. (2019). Attentive region embedding network for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. (pp. 9384–9393).

Yu, Yunlong, Ji, Zhong, Han, Jungong, & Zhang, Zhongfei (2020). 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).

Zhang, Ziming, & Saligrama, Venkatesh (2015). Zero-shot learning via semantic similarity embedding. In *Proceedings of the IEEE international conference on computer vision*. (pp. 4166–4174).

Zhang, Li, Xiang, Tao, & Gong, Shaogang (2017). Learning a deep embedding model for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. (pp. 2021–2030).

Zhu, Yizhe, Xie, Jianwen, Liu, Bingchen, & Elgammal, Ahmed (2019). Learning feature-to-feature translator by alternating back-propagation for generative zero-shot learning. In *Proceedings of the IEEE international conference on computer vision*. (pp. 9844–9854).