Imbalanced Continual Learning with Partitioning Reservoir Sampling

Chris Dongjoo Kim, Jinseo Jeong, and Gunhee Kim

Neural Processing Research Center, Seoul National University, Seoul, Korea
{cdjkim,jinseo}@vision.snu.ac.kr, gunhee@snu.ac.kr
http://vision.snu.ac.kr/projects/PRS

Abstract. Continual learning from a sequential stream of data is a crucial challenge for machine learning research. Most studies have been conducted on this topic under the single-label classification setting along with an assumption of balanced label distribution. This work expands this research horizon towards multi-label classification. In doing so, we identify unanticipated adversity innately existent in many multi-label datasets, the *long-tailed* distribution. We jointly address the two independently solved problems, Catastropic Forgetting and the long-tailed label distribution by first empirically showing a new challenge of destructive forgetting of the minority concepts on the tail. Then, we curate two benchmark datasets, *COCOseq* and *NUS-WIDEseq*, that allow the study of both *intra-* and *inter*-task imbalances. Lastly, we propose a new sampling strategy for replay-based approach named *Partitioning Reservoir Sampling* (PRS), which allows the model to maintain a balanced knowledge of both head and tail classes. We publicly release the dataset and the code in our project page.

Keywords: Imbalanced Learning, Continual Learning, Multi-Label Classification, Long-tailed distribution, Online Learning

1 Introduction

Sequential data streams are among the most natural forms of input for intelligent agents abiding the law of time. Recently, there has been much effort to better learn from these types of inputs, termed *continual learning* in machine learning research. Specifically, there have been many ventures into but not limited to single-label text classification [17], question answering [17], language instruction and translation [39], object detection [61,41], captioning [51] and even video representation learning [53,54]. Surprisingly, we have yet to see continual learning for multi-label classification, a more general and practical form of classification tasks since most real-world data are typically associated with several semantic concepts.

In order to study continual learning for multi-label classification, the first job would be to construct a research benchmark for it. We select two of the most popular multi-label datasets, MSCOCO [40] and NUS-WIDE [15], and tailor

them into a sequence of mutually exclusive tasks, *COCOseq* and *NUS-WIDEseq*. In the process, we recognize that large-scale multi-label datasets inevitably follow a *long-tailed* distribution where a small number of categories contain a large number of samples while most have only a small amount of samples. This naturally occurring phenomenon is widely observed in vision and language datasets [56,50,62], with a whole other branch of machine learning that has focused solely on this topic. Consequently, to effectively perform continual learning on multi-label data, two major obstacles should be overcome simultaneously: (i) the infamous *catastrophic forgetting* problem [49,55,23] and (ii) the long-tailed distribution problem [14,29,66,43], which we jointly address in this work.

We adopt the replay-based approach [44,28,3,60,58,38] to tackle continual learning, which explicitly stores the past experiences into a memory or a generative model, and rehearses them back with the new input samples. Although there also exists the prior-focused (*i.e.* regularization-based) [35,71,2] and expansion-based methods [59,70], the replay-based approaches have often shown superior results in terms of performance and memory efficiency. Specifically, the replay memory with *reservoir sampling* [64] has been a strong baseline, especially in the task-free continual setting [3,37] that does not require explicit task labels during the training nor test phase. It is an optimistic avenue of continual learning that we also undertake.

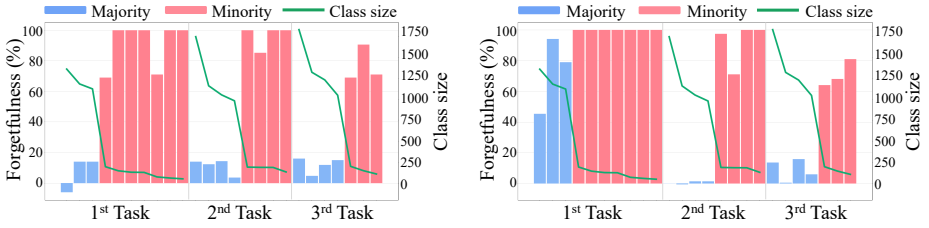
To conclude the introduction, we outline the contributions of this work:

I. To the best of our knowledge, this is the first work to tackle the continual learning for multi-label classification. To this end, we reveal that it is critical to correctly address the intra- and inter-task imbalances along with the prevailing catastrophic forgetting problem of continual learning.
II. For the study of this new problem, we extend the existing multi-label datasets into their continual versions called *COCOseq* and *NUS-WIDEseq*.
III. We propose a new replay method named *Partitioning Reservoir Sampling* (PRS) for continual learning in heterogeneous and long-tailed data streams. We discover that the key to success is to allocate a sufficient portion of memory to the moderate and minority classes to retain a balanced knowledge of present and past experiences.

2 Motivation: Fatal Forgetting on the Tail Classes

The long-tailed data distribution is both an enduring and pervasive problem in machine learning [33,29], as most real-world datasets are inherently imbalanced [56,50,74,62,52]. Wang *et al.* [66], for example, stated that minimizing the skew in the data distribution by collecting more examples in the tail classes is an arduous task and even if one manages to balance them along one dimension, they can become imbalanced in another.

We point out that the long-tailed distribution further aggravates the problem in continual learning, as a destructive amount of forgetting occurs on the tail classes. We illustrate this with experiments on two existing continual learning approaches: a prior-focused EWC [35] and a replay-based reservoir sampling [6,58,44].



(a) replay-based continual learning (CRS) (b) prior-focused continual learning (EWC)

Fig. 1. The forgetfulness of the majority and minority classes for two popular continual learning approaches over three sequential tasks. We test (a) EWC [35] and (b) replay-based reservoir sampling [6,58,44] with a shared output head and a memory size of 1000. We measure the forgetfulness using the metric proposed in [10] (Higher is more forgetful). The green line indicates the size of each class. More severe forgetting occurs for the minority classes in each task.

The experiments are carried out in an online setting on our *COCOseq* dataset, whose detail will be presented in section 5. Figure 1 shows the results. We plot the forgetting metric proposed in [10], which measures the difference between the peak performance and the performance at the end of the sequence. For illustrative purposes, we sort the classes per task in decreasing order of the number of classes. In both approaches, the minority (tail) classes experience more forgetting compared to the majority (head) classes. We observe that the imbalance of sample distribution in the memory causes this phenomenon in accordance with the input distribution, as we will further discuss in Figure 5.

3 Approach

The goal of this work is to overcome two inevitable obstacles of multi-label task-free continual learning: (i) catastrophic forgetting and (ii) long-tail input data distribution. Since we adopt the replay-based approach, we focus on a new sampling strategy to reserve past experiences into a fixed memory. We first clarify the problem (section 3.1), and discuss conventional reservoir sampling (section 3.2) and their fundamental limitations in this context (3.3). Finally, we propose our sampling method named *Partitioning Reservoir Sampling* (section 3.4).

3.1 Problem Formulation

We formulate our multi-label task-free continual learning as follows. The input is a data stream S, which consists of an unknown set of data points (x, y), where y is a multi-hot label vector representing k arbitrary number of classes. Except for the datapoint (x, y) that enters in an online-manner, no other information (*e.g.* the task boundaries or the number of classes) is available even during training. Given an input stream, the goal of the model is to allocate the fixed memory $\mathcal{M}$

with a size of m: $\sum_{i=1}^{u} m_i \leq m$, where m_i denotes the partitioned memory size for class c_i, and u is the unique number of classes observed so far at time t.

3.2 Conventional Reservoir Sampling

Conventional reservoir sampling (CRS) [64] maintains a fixed memory that uniformly samples from an input data stream. It is achieved by assigning a sampling probability of m/n to each datapoint where m is the memory size and n is the total number of samples seen so far. CRS is used as a standard sampling approach for task-free continual learning [58,11,57,44], since it does not require any prior information of the inputs but still attains an impressive performance [12]. However, its strength to uniformly represent the input distribution becomes its Achilles-heel in a long-tailed setting as the memory distribution also becomes long-tailed, leading to the realm of problems experienced in imbalanced training.

3.3 Fundamental Problems in Imbalanced Learning

Imbalanced data induce severe issues in learning that are primarily attributed to gradient dominance and under-representation of the minority [36,19,74,62].

(1) **Gradient dominance.** The imbalance in the minibatch causes the majority classes dominating the gradient updates, which ultimately lead to the neglect of the minority classes.

(2) **Under-representation of the minority.** Mainly due to the lack of data, the minority classes are much under-represented within the learned features relative to the majority [69,19]. We empirically confirm this in Figure 7, where the minority classes do not formulate a discernable pattern in the feature space but are sparsely distributed by conventional methods.

There have been data processing or algorithmic approaches to tackle these problems by promoting balance during training. Data processing methods such as oversampling or undersampling [5,7,52] explicitly simulate the input balance, while cost-sensitive approaches [31,66,16] adjust the update via regularizing the objective. More aggressively, there have also been directions that populate the minority samples via generation to avoid overfitting in the minority [13,46,20]. Most research in imbalanced learning shares the consensus that the *balance* during training is critical to success, which is the underlying emphasis on the design of our algorithm.

3.4 Partitioning Reservoir Sampling

Since a continual replay algorithm has no information about future input, the memory must maintain well-rounded knowledge in an online manner. To that end, we provide an online memory maintenance algorithm called *Partitioning Reservoir Sampling* (PRS) that consists of two fundamental operations: *partition* and *maintenance*. The PRS is overviewed in Figure 2 and Algorithm 1.

The Partition. During the training phase, the model only has access to the stream of data (x, y). While it is impractical to store all examples, caching

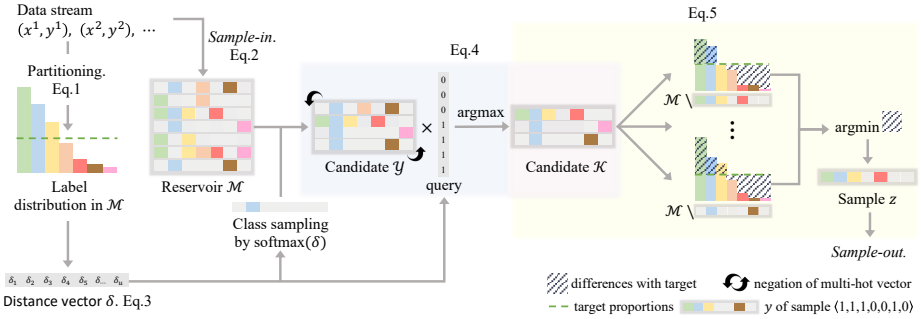


Fig. 2. Overview of Partitioning Reservoir Sampling. Based on the current data stream statistics, the target partition ratios are first obtained based on Eq. 1. We maintain the memory by iterating between the processes of *sample-in* and sample-out. The sample-in decides whether a new datapoint is stored into the memory or not, while the sample-out selects which example is removed from the memory. If the model allows to sample-in the datapoint, the algorithm traverses a process of candidate selection to sample-out an example by selecting the one that advances the memory towards the target partitions the most.

the running statistics is a more sensible alternative. Thus, the model uses the running class frequency to set the target proportion of classes in the memory. This is achieved by a variant of the proportional allocation [8,22,4]:

$$p_i = \frac{n_i^\rho}{\sum_j n_j^\rho}, \tag{1}$$

where ρ is a power of allocation, and n_i is the running frequency of class i. At $\rho = 0$, all classes are equally allocated, which may be the most favorable scenario for the minority as it shares the same amount of memory with the majority. At $\rho = 1$, classes are allocated proportionally to their frequencies, which is identical to the conventional sampling in section 3.2. ρ is chosen a value between 0 and 1 to compromise between the two extremes. For a given ρ, we can define the *target partition quota* for class i as $m_i = m \cdot p_i$ where p_i is defined by Eq. 1. Collectively, the target partition is represented as a vector $\mathbf{m} = [m_1, \cdots, m_u]$, whose sum is m. We will explore the effect of ρ in Figure 6.

The Maintenance. The goal of maintenance is to allow every *class* i (not every sample as in CRS) to have a fair chance of entering the memory according to the target partition m_i. To maintain a well-rounded knowledge of the past and present experiences, we iterate between the processes of *sample-in* and *sample-out*. The sample-in decides whether a new input datapoint is reserved into the memory or not, while the sample-out selects which example is removed from the memory when it becomes full and new samples continue to enter.

1) *Sample-in.* For an incoming datapoint, we assign a sampling probability s to be reserved in the memory. We compute s with two desiderata: (i) it needs

to comply with the target partition, and (ii) for better balancing, it is biased towards the minority classes with respect to the current running statistics.

$$s = \sum_{i \in \{i,..,u\}} \frac{m_i}{n_i} \cdot w_i, \quad \text{where } w_i = \frac{y_i e^{-n_i}}{\sum_{j=1} y_j e^{-n_j}} \tag{2}$$

where u is the unique number of classes observed, n_i is the running frequency of class i, and y_i is the datapoint's multi-hot vector value for class i. w_i is the normalized weight computed by the softmax of the negative running frequency of the classes. This formulation allows to bias w_i strongly towards the minority.

2) *Sample-out.* When the memory $\mathcal{M}$ is full and new samples continue to enter, we need to sample out an example from the memory while striving towards the target partition. The first order of matter would be to quantify the distance from the current memory partition to our target partition. To do so, we define a u-dimensional vector δ with each element as

$$\delta_i = l_i - p_i \cdot \sum_j l_j, \tag{3}$$

where l_i is the number of examples of class i in the memory and p_i is the partition ratio from Eq. 1. Note that we multiply p_i by $\sum_j l_j$ rather than the memory size m, due to the multiple labels on each datapoint.

In order to fulfill our objective (*i.e.* achieve the target partitions), we greedily select and remove the sample that best satisfies the following two desiderata: (i) include the classes that occupy more memory than their quota, *i.e.* $\delta_i > 0$ and (ii) exclude the classes that under-occupy or already satisfy the target, $\delta_i \leq 0$.

To this end, we devise a two-stage candidate selection process. For desideratum (i), we define a set of candidate sample $\mathcal{Y} \subset \mathcal{M}$ as follows. Among the classes with $\delta_i > 0$, we randomly sample a class with a probability of softmax(δ_i). This sampling is highly biased toward the class with the maximum δ_i value (*i.e.* the class to be reduced the most). We found this to be more robust in practice than considering multiple classes with $\delta_i > 0$. Then, $\mathcal{Y}$ contains all samples labeled with this selected class in the memory. For desideratum (ii), we define a u-dimensional indicator vector q where $q_i = 0$ if $\delta_i > 0$ and $q_i = 1$ otherwise. That is, q_i indicates the classes that do not over-occupy the memory. Finally, the set of candidate samples $\mathcal{K}$ is obtained by

$$\mathcal{K} = \{n^* | n^* = \arg\max_{n \in \mathcal{Y}} (\neg y^n \cdot q)\}, \tag{4}$$

where $\neg y^n$ is the negation of the multi-hot label vector of sample n (*i.e.* $0 \rightarrow 1$ and $1 \rightarrow 0$). That is, $\mathcal{K}$ is a subset of $\mathcal{Y}$ that does not contain sample(s) for the under-occupied (or already-satisfied) classes as possible. $\mathcal{K}$ may include multiple samples while the samples with fewer labels are more likely to be selected.

Finally, amongst $\mathcal{K}$, we select example z to be removed as it is the one that advances the memory towards the target partition the most:

$$z = \arg\min_{k \in \mathcal{K}} \sum_{i \in \{1,..,u\}} \left| \mathcal{C}_{ki} - p_i \cdot \sum_{l \in \{1,..,u\}} \mathcal{C}_{kl} \right|, \text{where } \mathcal{C}_{ki} = \sum_{n \in \mathcal{M} \setminus k} y_i^n. \tag{5}$$

Algorithm 1 Partitioning Reservoir Sampling Pseudo-code

Require: (i) data $(x_t, y_t), ..., (x_T, y_T)$, (ii) power param ρ, (iii) memory size m.
$\mathcal{M} = \{\}$ // memory
$\psi = \emptyset$ // running statistics
$u = 0$ // number of unique classes
for $t = 1$ **to** T **do**
  $update(\psi)$ // update running stats
  if $t \leq |m|$ **then**
    // fill memory
    $\mathcal{M}_u \leftarrow \mathcal{M}\{y_t\}$ // sub memory
    $\mathcal{M}_u \leftarrow \{x_t, y_t\} \cup \mathcal{M}_u$
  else
    // Partition
    $Partitioning(\mathcal{M}, \psi, q)$ // Eq. 1
    // Maintenance
    // class-indep reservoir sampling
    $sample\_in(\mathcal{M}_u, y_{t,u}, \psi)$ // Eq. 2
    if sample-in success **then**
      $sample\_out(\mathcal{M}, \psi, q)$ // Eq. 5
    end if
  end if
end for

$\mathcal{C}_{ki}$ is the current number of class i in the memory after the removal of sample k from memory $\mathcal{M}$, p_i is the partition ratio of class i from Eq. 1, and y_i^n is a binary value for class i of the label vector of sample n. Eq. 5 finds the sample z that minimizes the distance (defined in Eq. 3) towards the target partition before and after the removal of sample k.

4 Related Work

There have been three main branches in continual learning, which are regularization, expansion and replay methods. Here we focus on the replay-based approaches and present a more comprehensive survey in the Appendix.

Replay-based approaches. They explicitly maintain a fixed-sized memory in the form of generative weights or explicit data to rehearse it back to the model during training. Many recent works [32,6,58,44,11,12,57] employ a memory that reserves the data samples of prior classes in an offline setting. For example, GEM [44] uses the memory to constrain the gradient direction that prevents forgetting, and this idea becomes more efficient in AGEM [11]. Chaudhry *et al.* [12] explore tiny episodic memory, which shows improved overall performance when training repetitively from only a few examples. Riemer *et al.* [57] introduce a method that combines rehearsal with meta-learning to find the right balance between transfer and interference. Since our approach uses no prior knowledge other than the given input stream, it is orthogonally integrable with many aforementioned methods.

Online Sequential Learning. Recently, there have been some approaches to *online* continual learning where each training sample is seen only once. *ExStream* [28] is an online stream clustering reservoir method, but it requires prior knowledge about the number of classes to pre-allocate the memory. As new samples enter, the sub-memory is filled based on a distance measure and merged in the feature space when the memory is full. GSS [3] may be one of the most similar works to ours. It formulates the sample selection as a constraint

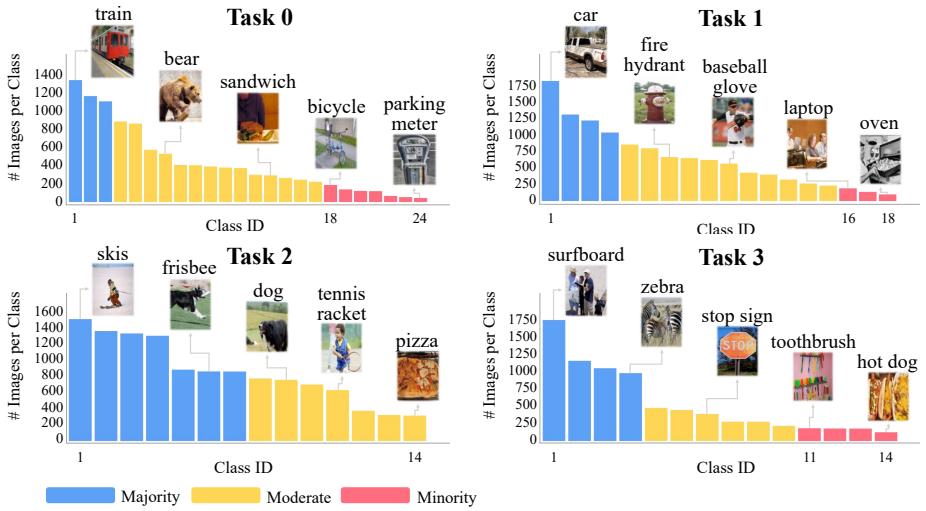


Fig. 3. Statistics of *COCOseq* dataset consisting of four tasks.

reduction problem, intending to select a fixed subset of constraints that best approximate the feasible region. They perform miniaturized MNIST experiments with different task sizes (*e.g.* 2000 instances for one task and 200 for the others). However, this setting is difficult to represent practical long-tailed or imbalanced problems, since only a single task is much larger than the other same-sized tasks.

Multi-label Classification. There have been many works handling the vital problem of multi-label classification [47,25]. Recently, recurrent approaches [65,67] and attention-based methods [73,26] are proposed to correlate the labels during predictions. Wei *et al.* [68] employ the prior task knowledge to perform graph-based learning that aids the correlation representation of multiple labels. While all the works in the past have focused on the offline multi-label classification, we take on its online task-free continual learning problem. Moreover, our approach is orthogonally applicable to these methods as we select some of them as the base model in our experiments.

5 The Multi-label Sequential Datasets

To study the proposed problem, we transform two multi-label classification datasets into their continual versions: *COCOseq* and *NUS-WIDEseq*. It is a non-trivial mission since the data must be split into tasks with exclusive class labels where each datapoint is associated with multiple labels.

5.1 The *COCOseq*

There have been two previous works that curate the MSCOCO dataset [40] for continual learning. Shmelkov *et al*. [61] select 20 out of 80 classes to create 2 tasks, each with 10 classes by grouping them based on alphabetical ordering. They also create an incremental version with 25 classes, where 15 classes are used for the model to obtain the base knowledge via normal batch training, and the other 10 classes are sequentially learned one at a time. These 10 classes are selected so that each image has only a *single* class label. Nguyen *et al*. [51] tailor MSCOCO for continual learning of captioning. They use 28 out of 80 classes and discarded all the images that contain multiple class labels. Similar to [61], they create 2 tasks where one task has 5 classes, and the other has 19 classes. Also, a sequential version is made using the 19 classes for base knowledge learning and the remaining 5 for incremental learning.

Different from previous works, we curate 4 tasks with multi-label images. To accurately measure the training performance on the intra-task imbalance, we make sure that the test set is *balanced*; the test set size per class is identical even though its training set size is imbalanced and long-tailed. This is a common practice in imbalanced classification benchmarks, including [66] that uses 40 test images per class in the SUN-LT dataset and the largest OLTR benchmark [43] that assigns 50 and 100 balanced test images per class for ImageNet-LT and Places-LT dataset, respectively. While referring to the Appendix for more details of dataset construction, we build a 4-way split MSCOCO dataset called *COCOseq* (Figure 3), consisting of 70 object classes with 35072 training and 6346 test data. The test set contains one-hundred images per class. Note that $6346 \neq 70 \times 100$ due to the multi-label property.

To the best of our knowledge, there is no strict consensus to divide the majority and minority classes in long-tailed datasets. For instance, Liu *et al*. [43] define the classes with more than 100 training examples as many-shot, 20-100 as medium-shot, and less than 20 as few-shot classes. Other works such as [27] and [72] define the minority classes as less than 100 or 200 samples in the training set, respectively. Accordingly, we define classes with less than 200 training examples as the minority classes, 200-900 as moderate and >900 as the majority.

5.2 The *NUS-WIDEseq*

We further curate a sequential dataset from NUS-WIDE [15], containing 6 mutually exclusive and increasingly difficult tasks. Its novelty lies in having both inter- and intra-task imbalance; the skewness exists not only within each task but amongst the tasks as well. More details can be found in the Appendix.

NUS-WIDE [15] is a raw web-crawled multi-label image dataset. It provides a human-annotated version with 150531 images of 81 labels[1]. However, the dataset by nature exhibits a very severe long-tail property. For instance, MSCOCO's top 20% classes are responsible for 50.7% of the total data, while NUS-WIDE's top

[1] The original number of images is 210832, but many URLs are no longer available.

20% surmount to 76.3% of the whole data. Since the original test set is highly long-tailed, we balance it for more accurate evaluation as done for *COCOseq*. Finally, *NUS-WIDEseq* contains 49 classes with 48724 training and 2367 test data with 50 samples per class.

6 Experiments

In our evaluation, we explore how effective our PRS is for both inter- and intra-task imbalances in task-free multi-label continual learning tasks compared to the state-of-the-art models. We also analyze the importance of a balanced memory from many aspects. The task that we solve is mostly close to but more difficult than the scenario of *class-incremental learning* [63] in that the task label is not available at training as well as at test time.

6.1 Experimental Design

Previous continual learning research has shown a high amount of disparity in evaluation. As we are the first to explore multi-label continual learning, we explicitly ground our experimental setting based on [21,1,63] as follows:

- *Cross-task resemblance*: Consecutive tasks in COCOseq and NUS-WIDEseq are partly correlated to contain neighboring domain concepts.
- *Shared output heads*: Since we solve multi-label classification without task labels, the level of difficulty of our task is comparable to using a shared output head for single-label classification.
- *No test-time task labels*: Our approach does not require explicit task labels during both training and test phase, often coined as *task-free continual learning* in [3,37].
- *More than two tasks*: COCOseq and NUS-WIDEseq contain four and six tasks, respectively.
- *Online learning*: The algorithm learns from a continuous stream of data without a separate offline batch training stage such as [3,37,28].

Base Models. In recent multi-label image classification [65,67,42,68,26], it is a common practice to fine-tune a pre-trained model to a target dataset. We thus employ ResNet101 [30] pre-trained on ImageNet [18] as our base classifier. Additionally, we test two multi-label classification approaches that *do not* require any prior information about the input to train: Recurrent Attention (RNN-Attention) [67] and the more recent Attention Consistency (AC) algorithm [26]. Due to its superior performance, we choose ResNet101 as the base model for the experiments in the main draft. We report the results of RNN-Attention and AC methods in the Appendix.

Evaluation Metrics. Following the convention of multi-label classification [65,73,24], we report the average overall F1 (O-F1), per-class F1 (C-F1) as well as the mAP. Additionally, we include the forgetting metric [10] to quantify the effectiveness of continual learning techniques. However, since this is a self-relative

Table 1. Results on *COCOseq* and *NUS-WIDEseq*. We report accuracy metrics for multi-label classification after the whole data stream is seen once. Similar to [43], the majority, moderate and minority are distinguished to accurately assess the long-tail performances. The memory size is fixed at 2000, with {0,3,1,2} task schedule for *COCOseq* and {3,1,0,5,4,2} for *NUS-WIDEseq*. The results are the means of five experiments except those of GSS-Greedy [3] which are the mean of three due to its computational complexity. The best and the second best methods are respectively marked in red and blue fonts, excluding the MULTITASK that is offline trained as the upper-bound. FORGET refers to the normalized forgetting measure of [10].

	MAJORITY			MODERATE			MINORITY			**Overall**		
COCOseq	C-F1	O-F1	MAP	C-F1	O-F1	MAP	C-F1	O-F1	MAP	C-F1	O-F1	MAP
MULTITASK [9]	72.9	70.9	77.3	53.2	51.4	55.0	12.7	13.6	24.2	51.2	52.1	53.9
FINETUNE	18.5	27.9	29.8	6.7	16.7	14.1	0.0	0.0	5.2	8.5	18.4	16.4
FORGET	100.0	100.0	65.8	100.0	100.0	73.5	100.0	100.0	67.4	100.0	100.0	70.1
EWC [35]	60.0	53.4	64.1	37.3	38.1	47.5	7.5	8.2	21.5	38.9	40.0	46.6
FORGET	24.2	24.0	0.8	35.1	33.9	3.0	56.3	56.1	9.0	32.8	32.0	3.2
CRS [64]	67.0	62.5	67.9	47.8	45.2	50.4	14.5	15.6	26.9	47.5	46.6	50.2
FORGET	15.0	13.6	8.9	32.8	32.0	15.6	55.58	54.92	23.2	32.2	30.1	15.3
GSS [3]	59.3	56.7	59.6	44.9	43.0	46.0	10.5	11.0	18.6	42.8	42.7	44.0
FORGET	20.2	18.8	10.3	36.4	35.3	13.6	67.4	68.4	26.1	35.1	35.3	13.6
EXSTREAM [28]	58.8	52.0	62.5	49.2	47.3	52.7	26.4	26.6	36.6	47.8	43.9	51.1
FORGET	41.8	40.2	17.7	33.9	32.9	14.7	47.0	33.4	15.5	40.5	39.6	16.2
PRS(OURS)	65.4	59.3	67.5	52.5	49.7	55.2	34.5	34.6	39.7	53.2	50.3	55.3
FORGET	22.0	21.7	8.5	27.2	26.8	11.5	26.2	26.3	10.5	25.6	25.2	10.2

	MAJORITY			MODERATE			MINORITY			**Overall**		
NUS-WIDEseq	C-F1	O-F1	MAP	C-F1	O-F1	MAP	C-F1	O-F1	MAP	C-F1	O-F1	MAP
MULTITASK [9]	33.7	30.8	32.8	29.3	28.7	28.9	9.7	11.8	25.8	24.6	24.9	28.4
FINETUNE	0.6	4.6	4.1	2.3	2.8	6.0	5.2	7.4	9.4	4.2	5.1	7.1
FORGET	100.0	100.0	47.3	100.0	100.0	39.3	100.0	100.0	44.6	100.0	100.0	44.4
EWC [35]	15.7	9.9	15.8	16.3	12.6	19.4	12.3	13.9	24.1	17.1	11.4	20.7
FORGET	18.4	15.4	7.8	64.6	63.7	7.3	63.4	63.4	4.8	36.4	31.5	7.3
CRS [64]	28.4	17.8	21.9	13.6	14.2	18.5	10.4	11.8	20.6	16.8	15.0	20.1
FORGET	33.6	29.2	14.7	67.8	66.7	18.1	96.5	96.2	20.3	61.5	57.8	18.7
GSS [3]	24.6	13.5	19.0	14.8	15.5	17.9	15.9	17.6	24.5	17.9	15.3	20.9
FORGET	46.8	43.9	16.6	59.6	58.3	11.7	82.8	82.0	18.6	54.8	49.8	13.0
EXSTREAM [28]	15.6	9.2	15.3	12.4	12.8	17.6	24.6	24.1	26.7	18.7	16.0	21.0
FORGET	80.7	77.6	24.0	81.0	80.6	23.3	77.2	76.7	21.8	81.0	79.3	23.4
PRS(OURS)	26.7	17.9	21.2	19.2	19.3	21.5	27.5	26.8	31.0	24.8	21.7	25.5
FORGET	45.8	43.0	15.7	59.0	58.4	13.4	60.6	60.3	15.5	55.3	53.5	13.9

metric on the best past and present performance of the method, comparisons between different methods could be misleading (*e.g.* if a model performs poorly throughout training, small forgetting metric values can be observed as it has little information to forget from the beginning). It is the reason for the absence of color for the best models with respect to this metric in the tables. In the Appendix, we also report the overall precision (O-P), recall (O-R), per-class precision (C-P) and recall (C-R) metrics.

Baselines. We compare our approach with six baselines including four state-of-the-art continual learning methods: EWC [35], CRS [64], GSS-Greedy [3] and ExStream [28]. In addition, the Multitask [9] can be regarded as an upper-

Table 2. Results according to memory sizes and schedule permutations on *COCOseq*. We fix the memory size of 2000 for schedule experiments and the schedule of {0,3,1,2} for memory experiments. Refer to Table 1 for the nomenclatures.

COCOseq	Overall C-F1	Overall O-F1	Overall MAP	Overall C-F1	Overall O-F1	Overall MAP
	SCHEDULE: 0, 1, 3, 2			MEMORY: 1000		
CRS[64]	49.2	46.9	50.8	44.2	41.6	47.1
GSS[3]	42.1	41.4	44.0	40.6	39.0	42.1
EXSTREAM[28]	47.3	42.9	50.5	41.6	37.6	47.0
PRS(OURS)	52.6	50.5	55.2	47.4	43.4	51.2
	SCHEDULE: 2, 3, 0, 1			MEMORY: 2000		
CRS[64]	45.4	44.3	48.2	47.5	46.6	50.2
GSS[3]	33.5	33.9	38.5	43.2	43.0	44.4
EXSTREAM[28]	41.6	35.2	45.7	47.8	43.9	51.1
PRS(OURS)	50.4	47.7	53.1	53.2	50.3	55.3
	SCHEDULE: 3, 1, 0, 2			MEMORY: 3000		
CRS[64]	47.7	45.9	49.7	49.4	48.6	51.1
GSS[3]	37.8	38.8	40.2	42.2	43.0	44.3
EXSTREAM[28]	45.4	41.8	49.2	49.7	46.8	52.2
PRS(OURS)	51.4	48.9	54.1	54.9	53.4	56.7

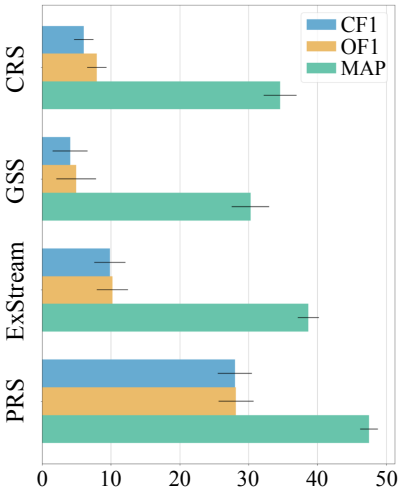


Fig. 4. Inter-task imbalance analysis on *NUS-WIDEseq*. We compare the performance for the smallest Task 3, for which our PRS robustly outperforms all the baselines.

bound performance as it is learned offline with minibatch training for a single epoch. The Finetune performs online training without any continual learning technique, and thus it can be regarded as a lower-bound performance. For training EWC, we fix the ResNet up to the penultimate layer in order to obtain sensible results; otherwise, it works poorly. More details for baselines are presented in the Appendix.

We use a fixed online input batch size of 10 and a replay-batch size of 10 in accordance with [3]. We use Adam [34] optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e-4$, and finetune all the layers unless stated otherwise. Furthermore, we set ρ between the range of [-0.2, 0.2], and fix the memory size to 2000 (as done in [48]), which are 5.7% and 4.7% of the overall training data for *COCOseq* and *NUS-WIDEseq*, respectively.

6.2 Results

Table 1 compares continual learning performance between our PRS method and baselines on *COCOseq* and *NUS-WIDEseq*. In all comparable metrics of C-F1, O-F1 and mAP, PRS outperforms CRS [57], GSS [3] and even ExStream [28] that uses prior task information to pre-allocate the memory.

Schedule and Memory Permutations. Table 2 compares the robustness of PRS through random permutations of task schedule as well as different memory sizes. Interestingly, the performances of CRS, GSS and ExStream fluctuates depending on the permuted schedule, while our PRS is comparatively robust

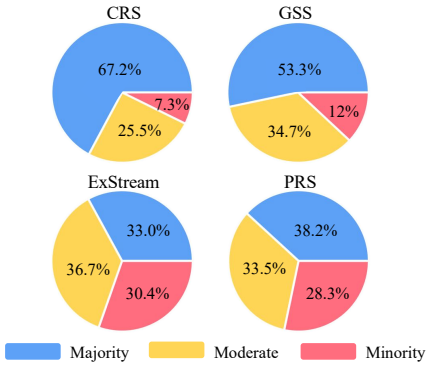


Fig. 5. The resulting memory distribution of the *COCOseq* tests in Table 1. ExStream [28] is the only *task-aware* method that knows the task distribution beforehand.

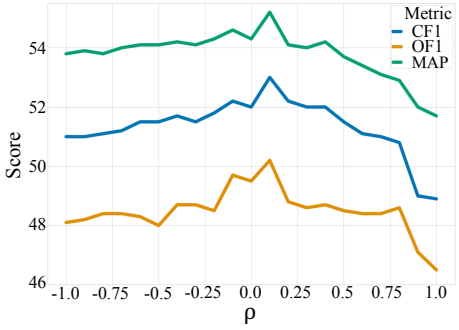


Fig. 6. Performance of PRS with different ρ in the range of [-1, +1] on *COCOseq*. All results are the averages of 5 different random seeds.

thanks to the balanced emphasis on all the learned classes. Moreover, PRS outperforms all the baselines with multiple memory sizes of $1000, 2000, 3000$.

Intra- and Inter-task Imbalance. Table 1 shows that PRS is competitive on the majority classes (*e.g.* marked in blue as the runner-up) and performs the best on both moderate and minority classes, showing its compelling robustness for the intra-task imbalances. Furthermore, Figure 4 validates the robustness of PRS in the inter-task imbalance setting. As shown in Fig. 1 of the Appendix, tasks of NUS-WIDEseq are *imbalanced* in that the smallest Task 3 is 9.6 times smaller than that of the largest Task 1. Figure 4 compares the performances of all methods for the minority Task 3, for which PRS performs overwhelmingly better than the other baselines in all the metrics.

Memory Distribution After Training. Figure 5 compares the normalized memory distribution of the experiments in Table 1. CRS dominantly uses the memory for the majority classes while reserving only a small portion for the minority. This explains why CRS may perform better than PRS for the majority in Table 1, while sacrificing performance largely for the moderate and minority classes. On the other hand, GSS saves much more samples for the moderate classes relative to CRS, but still fails to maintain a sufficient number of samples for the minority. Note that Exstream balances the memory using *prior task information*. However, due to its clustering scheme via feature merging to maintain the memory, it is difficult to obtain representative clusters, especially when handling complex datasets with multi-labels. Importantly, PRS can balance the memory for all classes without any auxiliary task information.

Power of Allocation ρ. Figure 6 shows the performance variation according to different ρ. It confirms that a balance of memory is vital for the performance even when the input stream is highly imbalanced. Notice, as ρ moves away from the vicinity of balance, the performance gradually declines in all metrics.

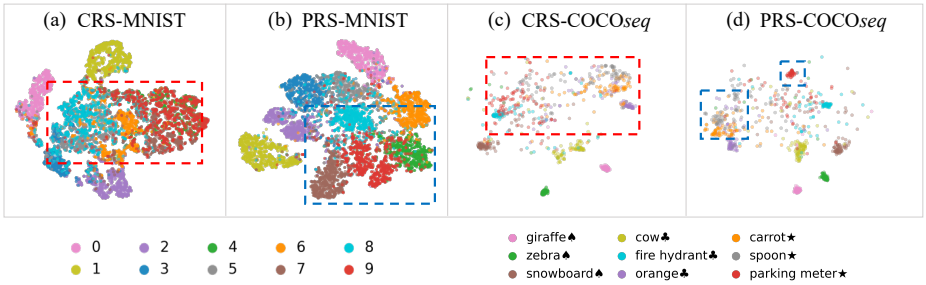


Fig. 7. t-SNE feature projection of CNN backbones trained by CRS and PRS for test samples of (a)–(b) MNIST and (c)–(d) COCOseq. We use the penultimate features of a 2-layer feedforward for MNIST and ResNet101 for COCOseq. As with the single-label experiments in the Appendix, we curate a sequential MNIST that follows a Pareto distribution [56] with a power value α=0.6, which becomes increasingly long-tailed from 0 to 9. For COCOseq, we use the symbols of {Minority: ★, Moderate: ♣, Majority: ♠}. We emphasize that PRS represents the minority classes (in the blue box) much more discriminatively than the corresponding class features (in the red box) for CRS.

Feature Analysis Figure 7 shows the feature projections of CNN backbones trained by CRS and PRS for test samples of MNIST and COCOseq using t-SNE [45]. PRS can represent the minority classes more discriminatively than CRS on both single-label MNIST and multi-label COCOseq experiments.

In the Appendix, we include more experimental results, including analysis on the memory gradients and performance on single-label classification and many more.

7 Conclusion

This work explored a novel problem of multi-label continual learning, which naturally requires the model to learn from imbalanced data streams. We contributed two datasets and an effective memory maintenance algorithm, called *Partitioning Reservoir Sampling* to tackle this new challenge. Our results showed the importance of maintaining a well-rounded knowledge through balanced replay memory. As a future direction of research, the ability to learn online while automatically tuning the target partitions would be an exciting avenue to explore.

Acknowledgements. We express our gratitude for the helpful comments on the manuscript by Soochan Lee, Junsoo Ha and Hyunwoo Kim. This work was supported by Samsung Advanced Institute of Technology, Institute of Information & communications Technology Planning & Evaluation (IITP) grant (No.2019-0-01082, SW StarLab) and the international cooperation program by the NRF of Korea (NRF-2018K2A9A2A11080927). Gunhee Kim is the corresponding author.

References

1. Aljundi, R.: Continual Learning in Neural Networks. Ph.D. thesis, Department of Electrical Engineering, KU Leuven (2019)
2. Aljundi, R., Marcus, R., Tuytelaars, T.: Selfless sequential learning. In: ICLR (2019)
3. Aljundi, R., Lin, M., Goujaud, B., Bengio, Y.: Gradient based sample selection for online continual learning. In: NeurIPS (2019)
4. Bankier, M.: Power Allocations: Determining Sample Sizes for Subnational Areas. The American Statician (1988)
5. Batista, G., Prati, R., Monard, M.: A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explor **6**, 20–29 (2004)
6. Brahma, P.P., Othon, A.: Subset replay based continual learning for scalable improvement of autonomous systems. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2018)
7. Buda, M., Maki, A., Mazuorowski, M.: A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks **106**, 249–259 (2018)
8. Carroll, J.: Allocation of a Sample Between States. Australian Bureau of Census and Statistics (1970)
9. Caruaca, R.: Multitask learning. Machine Learning **28**, 41–75 (1997)
10. Chaudhry, A., Dokania, P.K., Ajanthan, T., Torr, P.H.: Riemannian walk for incremental learning: Understanding forgetting and intransigence. In: ECCV (2018)
11. Chaudhry, A., Ranzato, M., Rohrbach, M., Elhoseiny, M.: Efficient lifelong learning with a-gem. In: ICLR (2019)
12. Chaudhry, A., Rohrbach, M., Elhoseiny, M., Ajanthan, T., Dokania, P.K., Torr, P.H., Ranzato, M.: On tiny episodic memories in continual learning. arXiv preprint arXiv:1902.10486v4 (2019)
13. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)
14. Chen, C., Liaw, A., Breiman, L., et al.: Using random forest to learn imbalanced data. University of California, Berkeley **110**(1-12), 24 (2004)
15. Chua, T.S., Tang, J., Hong, R., Li, H., luo, Z., Zheng, Y.: Nus-wide: a real-world web image database from national university of singapore. In: ACM (2009)
16. Cui, Y., Jia, M., Lin, T., Song, Y., Belongie, S.: Class-balanced loss based on effective number of samples. In: CVPR (2019)
17. d'Autume, C., Ruder, S., Kong, L., Yogatama, D.: Episodic memory in lifelong language learning. In: NeurIPS (2019)
18. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Li, F.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
19. Dong, Q., Gong, S., Zhu, X.: Imbalanced deep learning by minority class incremental rectification. TPAMI **41**, 1367–1381 (2019)
20. Douzas, G., Bacao, F.: Effective data generation for imbalanced learning using conditional generative adversarial networks. Expert Systems with applications **91**, 464–471 (2018)
21. Farquhar, S., Gal, Y.: Towards robust evaluations of continual learning. arXiv preprint arXiv:1805.09733 (2019)
22. Fellegi, I.P.: Should the Census Counts Be Adjusted for Allocation Purposes?-Equity Considerations. Current Topics in Survey Sampling pp. 47–76 (1981)
23. French, R.: Catastrophic forgetting in connectionist networks. Trends in Cognitive Sciences **3**(4), 128–135 (1999)

24. Ge, W., Yang, S., Yu, Y.: Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning. In: CVPR (2018)
25. Guillaumin, M., Mensink, T., Verbeek, J., Schmid, C.: Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In: ICCV (2009)
26. Guo, H., Zheng, K., Fan, X., Yu, H., Wang, S.: Visual Attention Consistency under Image Transforms for Multi-Label Image Classification. In: CVPR (2019)
27. Han, X., Yu, P., Liu, Z., Sun, M., Li, P.: Hierarchical relation extraction with coarse-to-fine grained attention. In: EMNLP (2018)
28. Hayes, T.L., Cahill, N.D., Kanan, C.: Memory efficient experience replay for streaming learning. In: 2019 International Conference on Robotics and Automation (ICRA) (2019)
29. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering **9**, 1263–1284 (2008)
30. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
31. Huang, C., Li, Y., Change Loy, C., Tang, X.: Learning deep representation for imbalanced classification. In: CVPR (2016)
32. Isele, D., Cosgun, A.: Selective experience replay for lifelong learning. In: AAAI (2018)
33. Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. Intelligent Data Analysis **6**, 429–449 (2002)
34. Kingma, D., Ba, J.: Adam: A Method for Stochastic Optimization. In: ICLR (2015)
35. 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., Hadsell, R.: Overcoming catastrophic forgetting in neural networks. In: Proceedings of the National Academy of Sciences (2017)
36. Krawczyk, B.: Learning from imbalanced data:open challenges and future directions. Progress in Artificial Intelligence **5**, 221–232 (2016)
37. Lee, S., Ha, J., Zhang, D., Kim, G.: A neural dirichlet process mixture model for task-free continual learning. In: ICLR (2020)
38. Lesort, T., Gepperth, A., Stoian, A., Filliat, D.: Marginal replay vs conditional replay for continual learning. In: IJCANN (2019)
39. Li, Y., Zhao, L., Church, K., Elhoseiny, M.: Compositional continual language learning. In: ICLR (2020)
40. Lin, T., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, L.C.: Microsoft COCO: Common objects in Context. In: ECCV (2014)
41. Liu, Y., Cong, Y., Sun, G.: L3doc: Lifelong 3d object classification. arXiv preprint arXiv:1912.06135 (2019)
42. Liu, Y., Sheng, L., Shao, J., Yan, J., Xiang, S., Pan, C.: Multi-Label Image Classification via Knowledge Distillation from Weakly-Supervised Detection. In: ACM (2018)
43. Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., Yu, S.X.: Large-scale long-tailed recognition in an open world. In: CVPR (2019)
44. Lopez-Paz, D., Ranzato, M.: Gradient episodic memory for continual learning. In: NeurIPS (2017)
45. van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research **9**, 2579–2605 (2008)
46. Maciejewski, T., Stefanowski, J.: Local neighbourhood extension of smote for mining imbalanced data. In: CIDM (2011)

47. Makadia, A., Pavlovic, V., Kumar, S.: A new baseline for image annotation. In: ECCV (2008)
48. Maltoni, D., Lomonaco, V.: Continuous learning in single-incremental-task scenarios. Elsevier Neural Networks Journal **116**, 56–73 (2019)
49. McCloskey, M., Cohen, N.J.: Catastrophic interference in conncectionist networks. Psychology of learning and motivation **24**, 109–265 (1989)
50. Newman, M.: Power laws, pareto distributions and zipf's law. Contemporary Physics **46**, 323–351 (2005)
51. Nguyen, G., Jun, T.J., Tran, T., Kim, D.: Contcap: A comprehensive framework for continual image captioning. arXiv preprint arXiv:1909.08745 (2019)
52. Ouyang, W., Wang, X., Zhang, C., Yang, X.: Factors in finetuning deep model for object detection with long-tail distribution. In: CVPR (2016)
53. Parisi, G.I., Tani, J., Weber, C., Wermter, S.: Lifelong learning of human actions with deep neural network self-organization. In: Neural Networks (2017)
54. Parisi, G.I., Tani, J., Weber, C., Wermter, S.: Lifelong learning of spatiotemporal representations with dual-memory recurrent self-organization. Frontiers in Neurorobotics **12** (Nov 2018)
55. Ratcliff, R.: Conncectionist models of recognition memory: Constraints imposed by learning and forgetting functions. Pscyhological review **97**(2), 285–308 (1990)
56. Reed, W.J.: The pareto, zipf and other power laws. Economic letters **74**, 15–19 (2001)
57. Riemer, M., Cases, I., Ajemian, R., Liu, M., Rish, I., Tu, Y., Tesauro, G.: Learning to learn without forgetting by maximizing transfer and minimizing interference. In: ICLR (2019)
58. Rolnick, D., Ahuja, A., Schwarz, J., Lillicrap, T.P., Wayne, G.: Experience replay for continual learning. In: NeurIPS (2019)
59. Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. arXiv preprint arXiv:1606.04671 (2016)
60. Shin, H., Lee, J.K., Kim, J., Kim, J.: Continual learning with deep generative replay. In: NeurIPS (2017)
61. Shmelkov, K., Schmid, C., Alahari, K.: Incremental learning of object detectors without catastrophic forgetting. In: ICCV (2017)
62. Van Horn, G., Perona, P.: The devil is in the tails: Fine-grained classification in the wild. arXiv preprint arXiv:1709.01450 (2017)
63. van de Ven, G.M., Andreas, S.T.: Three scenarios for continual learning. In: NeurIPS Continual Learning workshop (2019)
64. Vitter, J.S.: Random sampling with a reservoir. ACM Transactions on Mathematical Software (TOMS) **11**(1), 37–57 (1985)
65. Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., Xu, W.: CNN-RNN: A Unified Framework for Multi-label Image Classification. In: CVPR (2016)
66. Wang, Y., Ramana, D., Hebert, M.: Lerning to model the tail. In: NeurIPS (2017)
67. Wang, Z., Chen, T., Li, G., Xu, R., Lin, L.: Multi-label Image Recognition by Recurrently Discovering Attentional Regions. In: ICCV (2017)
68. Wei, Z.M.C.X.S., Wang, P., Guo, Y.: Multi-Label Image Recognition with Graph Convolutional Networks. In: CVPR (2019)
69. Yin, X., Yu, X., Sohn, K., Liu, X., Chandraker, M.: Feature transfer learning for deep face recognition with under-represented data. In: CVPR (2019)
70. Yoon, J., Yang, E., Lee, J., Hwang, S.J.: Lifelong learning with dynamically expandable networks. In: ICLR (2018)

71. Zenke, F., Poole, B., Ganguli, S.: Continual learning through syanptic intelligence. In: ICML (2017)
72. Zhang, N., Deng, S., Sun, Z., Wang, G., Chen, X., Zhang, W., Chen, H.: Long-tail relation extraction via knowledge graph embeddings and. In: NAACL (2019)
73. Zhu, F., Li, H., Ouyang, W., Yu, N., Wang, X.: Learning spatial regularization with imagelevel supervisions for multi-label image classification. In: CVPR (2017)
74. Zhu, X., Anguelov, D., Ramanan, D.: Capturing long-tail distributions of object subcategories. In: CVPR (2014)