Rainbow Memory: Continual Learning with a Memory of Diverse Samples

Jihwan Bang[1,*] Heesu Kim[2,3,*] YoungJoon Yoo[2,3] Jung-Woo Ha[2,3] Jonghyun Choi[4,†]
Search Solutions, Inc[1] NAVER CLOVA[2] NAVER AI Lab[3] GIST[4]
{jihwan.bang,heesu.kim89,youngjoon.yoo,jungwoo.ha}@navercorp.com, jhc@gist.ac.kr

Abstract

Continual learning is a realistic learning scenario for AI models. Prevalent scenario of continual learning, however, assumes disjoint sets of classes as tasks and is less realistic rather artificial. Instead, we focus on 'blurry' task boundary; where tasks shares classes and is more realistic and practical. To address such task, we argue the importance of diversity of samples in an episodic memory. To enhance the sample diversity in the memory, we propose a novel memory management strategy based on per-sample classification uncertainty and data augmentation, named Rainbow Memory (RM). With extensive empirical validations on MNIST, CIFAR10, CIFAR100, and ImageNet datasets, we show that the proposed method significantly improves the accuracy in blurry continual learning setups, outperforming state of the arts by large margins despite its simplicity. Code and data splits will be available in https://github.com/clovaai/rainbow-memory.

1. Introduction

Continual learning (CL) or class incremental learning (CIL) is known to particularly suffer from the catastrophic forgetting with respect to model generalization, due to inaccessibility to the data of previous tasks. It is because the class distributions of each task continuously change given a task stream. This makes continual learning particularly challenging; most AI models suffer from under real-world application scenarios across various domains [38, 20, 30]. To address the issue of changing data distribution for continual learning, there are many approaches proposed such as momentum matching [29], sample generation [42, 46, 24, 43], regularization on parameters [27, 5], and sampling-based memory management [38, 39].

However, they are mostly evaluated in a rather artificial task setup of *disjoint*, where tasks do not share the classes [37]. For real-world applications, we consider a more realistic and practical setting of *blurry-CIL* where the classes shared across the tasks [38] (illustrated at the top of Figure 1). The blurry-CIL setup requires that (1) each task is given sequentially as a stream, (2) the majority (assigned) classes of tasks differ from each other, and (3) a model can leverage only a very small portion of data of previous tasks. For instance, suppose an e-commerce service that categorizes new items with their images taken by a seller. For each category, the number of newly registered items conspicuously depend on various factors such as season and transient event. The popularity period of items varies according to their characteristics as shown in Figure 2; *e.g.*, swimming suits are prevalent in summer and padding jumper are much more registered in winter.

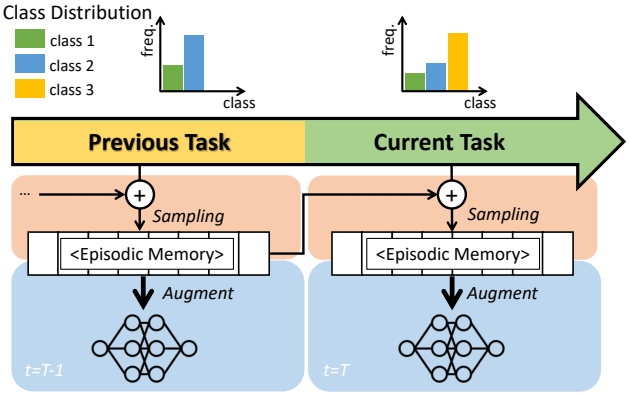


Figure 1: Blurry-CIL (class incremental learning) setup (top) and overview of our proposed approach (bottom). In the blurry-CIL, the tasks share classes, contrary to conventional disjoint-CIL. Proposed memory management strategy updates an episodic memory with samples of the current task to keep diverse exemplars in the memory. Data augmentation (DA) further enhances the diversity of the kept exemplars.

In recent literature, the methods storing a small portion of old data have shown promising results in preserving the information of old classes when training new classes for the blurry-CIL setup [38], thus alleviating catastrophic forgetting [16]. This strategy naturally raises the question: *what is the optimal strategies to manage the memory?* Since the number of stored samples is much smaller than that of the

* indicates equal contribution. † indicates corresponding author.

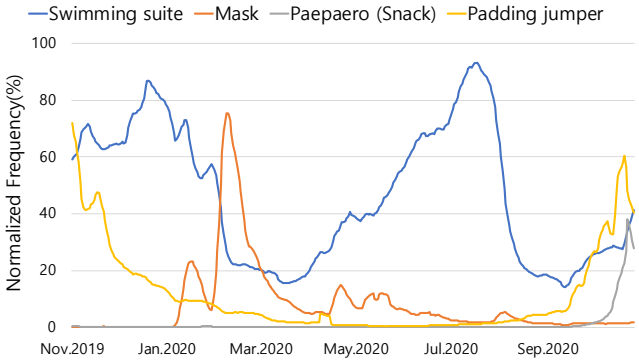


Figure 2: Popularity changes of four items including swimming suite, snack, mask, and padding jumper during one year in a real-world e-commerce service. Each item has its own popularity period and this phenomena is more similar to blurry-CIL than disjoint-CIL because most item categories do not disjointly appear in real-world applications.

incoming new-class, the samples in the memory would incur either overfitting or be ignored during training due to its small size compared to that of samples of incoming tasks. As a straightforward solution, if we gradually increase the memory size when the samples are incoming, the problem-setting fails to hold an important resource constraint of the CIL; a limited fixed memory requirement. Therefore, we need a strategy to maintain sufficient information of the old class with a small number of samples.

To address this problem, we investigate two factors for better continual learning on the newly defined *blurry-CIL* setup; *sampling* for the memory and *augmenting* the data in the memory. First, we propose a *perturbation-induced uncertainty* to select samples for the memory by measuring the per-sample robustness against the perturbations. To measure the uncertainty, we define a prior distribution that draws the perturbed samples and approximates the *robustness* (*i.e.*, inverse of uncertainty) described as a likelihood function in a Bayesian formulation. We fill the memory slots with the samples drawn from the distribution corresponding to the robustness. We show that the diversity-induced memory by sampling both perturbation-robust and fragile data helps the models to preserve discriminative boundary for each class.

Second, we investigate the effect of the diversity acquired by data augmentation in the blurry-CIL. In particular, label mixing-based data augmentation, such as CutMix [49], projects the input samples into a more complex dimension by mixing the image-label of multiple data samples randomly and has reported notable successes in various recognition tasks [47, 26]. It provides additionally rich diversity of stored samples in the episodic memory. Along with the label mixing augmentation, we exploit the effects of composition of multiple data augmentations for enhancing the diversity, benefiting from conventional methods such as flipping, shearing, or color jittering and recent automated data augmentation researches [9, 10, 32]. Incorporating the two proposals, we name our method as *Rainbow Memory* or *RM* for short.

Our RM is mainly evaluated in blurry-CIL setup on MNIST, CIFAR10, CIFAR100 and ImageNet datasets, compared with various standard CIL methods. The extensive experimental validations show that our approach effectively addresses blurry-CIL, outperforming state-of-the art baselines with significant margins (over 14%p and 9% of accuracy on all datasets evaluated). In addition, our method comparably performs to the other methods in disjoint-CIL set-up even if it is designed for blurry-CIL setup.

We summarize the contributions as follows:

- We propose a new diversity-aware sampling method for effectively managing the memory with limited capacity by leveraging classification uncertainty.
- We propose to augment the samples in the memory to further enhance the diversity of the samples.
- Our RM outperforms previous methods in blurry-CIL setup by large margins.
- We release the source code of RM and the evaluation protocol including the task splits of blurry-CIL for future research in this avenue.

2. Related Work

Class Incremental Learning Setups. Among many scenarios of continual learning, summarized in [43], we are particularly interested in class-incremental learning (CIL) scenario with no task identity is given at the inference [15]. There have been many proposals that can be roughly categorized into (1) rehearsal-based approaches [6, 4, 39], where episodic memory stores a few exemplars of old tasks, then the exemplars will be replayed in the future task, and (2) regularization-based approaches [28, 50, 33, 31, 29, 36], where no samples of old tasks is stored, but exploit the information of old tasks implicitly remained in the parameters of models. As rehearsal-based approaches generally have shown the better performance in CIL [38], we propose to improve memory management and exploit the insufficient information in an episodic memory, presuming the existence of such memory.

Class-incremental learning usually refers to a sequential learning paradigm with disjoint set of tasks [39, 4, 15]. However, recent studies [1, 38] introduce a setup containing blurry and continuous stream of tasks, which is more realistic as many real-world tasks are seldom given in a disjoint manner. Another setup is whether CIL allows the temporary buffer for storing incoming samples of a current task or not during model training, each of which is called *offline* and *online*, respectively. Many previous works have been evaluated either of online [13, 1, 23] or offline [48, 39, 5, 4] setup, while GDumb [38] reports on both of setups. Basically, online is more difficult but more practical, so we focus

on online to report more practical results. Instead, we investigate the importance of memory management and propose effective memory update algorithm.

Class Imbalance. Rehearsal-based approaches have reported severe catastrophic forgetting due to the class-imbalance of exemplars [48]. This makes models vulnerable to the most frequent classes in episodic memory. To address the catastrophic forgetting problem, GEM [35], MER [40], and GSS [1] propose to update the weights using gradient information so that the models get knowledge from prior task, and BiC [48] proposes adding a simple layer at the end of model to calibrate the bias. Very recently, MEGA [17] proposes a loss balancing approach mixing loss of previous and current classes to relieve the forgetting. HAL [7] proposes a way to utilize the most destructive samples in the past tasks as anchor points to address the forgetting problem, and CAL [2] proposes an approach keeping additional information by storing intermediate activations, in addition to the raw images. However, those approaches overlook the importance of memory management and normally adopt simple random sampling [17, 2] or reservoir sampling [40] or ring-buffer sampling [7].

Episodic Memory Management. There are a number of strategies proposed in the literature [37]. Interestingly, many proposals show marginal accuracy improvement over the uniform random sampling despite the computational complexity [5, 4, 39]. These methods include herding selection [45], a discriminative samplings [34] and entropy based samples [6]. The herding selection chooses the samples proportional to a histogram of each sample's distance to the class mean. The discriminative sampling chooses the samples that define decision boundaries. The entropy based sampling method chooses the samples by the entropy of their softmax distribution in the output layer.

To obtain the representative and discriminative exemplars, Liu *et al*. proposes a complex but effective sampling method guaranteeing that the exemplars well represents the mean and boundary of each class distribution [34]. Also, Borsos *et al*. propose a coreset generation method for the representative memory using cardinality-constrained bi-level optimization [3]. and Cong *et al*. propose a GAN based memory which they can perturb styles of remembered samples for incremental learning [8]. These recently published works address the quality of the samples stored in the memory, they are either computationally expensive or difficult to train a sample generator for the memory [3].

Other than sampling, there are works addressing the episodic memory. Generative models are employed to generate past task samples [42, 41, 46, 21] instead of sampling. The generation strategy is an active research topic and shows promising results in relatively straightforward experimental validation (*e.g*., on MNIST and SVHN). But on these datasets, sampling from the uniform distribution already achieves saturated accuracy [6] and there is no promising results reported in challenging datasets (*e.g*., ImageNet) yet. Hayes *et al*. propose to replay 'compressed memory' to increase the memory utilization [18]. Iscen *et al*. propose to reduce the dimension of stored features for efficiency [22]. Fini *et al*. propose a batch-level distillation (BLD) method to increase the memory efficiency in an online setting which has an extreme memory constraint [13]. Unlike these works addressing the sampling efficiency, we focus on the quality of the stored samples in the memory.

3. Class Incremental Learning Setups

We can formulate CIL setups as follows:

$$
\begin{aligned}
C &= \{c_1, c_2, \ldots, c_N\}, \\
T_t &= \{c \mid \psi(c) = t\}, \\
\mathfrak{D}_c^C &= \{x_1^c, x_2^c, \ldots, x_{M_c}^c\}, \\
\mathfrak{D}_t^T &= \{\mathfrak{D}_c^C \mid c \in T_t\},
\end{aligned}
$$

where C denotes a set of all classes, T_t denotes a class-subset assigned to each task t, which is determined by a stochastic assign function, $\psi(c)$ returning an assigned task for a given class c, and $\mathfrak{D}_c^C$ and $\mathfrak{D}_t^T$ represent a set of samples populating class c and task t sample space, respectively. Note that N is not known and not even bounded in real-world scenario and M_c can be either of equal or not among classes (c) according to a problem definition.

We now formulate either blurry or disjoint CIL setups by intersecting $\mathfrak{D}_t^T$'s or not.

$$
\begin{aligned}
\text{disjoint-CIL} &\Rightarrow \bigcap T_t = \varnothing, \\
\text{blurry-CIL} &\Rightarrow \bigcap T_t \neq \varnothing.
\end{aligned}
$$

The disjoint-CIL setup exaggerates the catastrophic forgetting since it never exposes seen classes in successive tasks, but it is deviated from the real-world where new classes do not show up exclusively. Conversely, blurry-CIL setup makes the task boundaries faint in a way that each task contains small number of classes also present in the other tasks. Approaches are evaluated in various perspectives including forgetting and intransigence [5] under a continuously changing class balance setup [38].

4. Approach

To effectively address the blurry-CIL with an episodic memory, we propose a memory management strategy that enhances diversity of samples to cover the distribution of the class by sampling a diverse set of samples which may preserve the boundary of a class distribution. We further enhance the diversity of the samples by data augmentation.

4.1. Diversity-Aware Memory Update

We argue that the exemplars which are selected to be stored in the memory should be not only representative for their corresponding class but also discriminative to the other classes. To choose such samples, we argue that the samples that are near the classification boundary are the most discriminative and the samples that are close to the center of the distribution is the most representative. To satisfy both characteristics, we propose to sample the exemplars that are *diverse* in the feature space.

To secure the diversity, we need to estimate the relative locations of each sample in class-discriminative feature space. But it is computationally expensive to compute the relative locations of the features as it requires to compute sample-to-sample distances ($O(N^2)$). Instead, we propose to estimate the relative location by *uncertainty* of a sample estimated by the classification model, *i.e.*, we assume that the more certain samples for the model will be located closer to the center of the class distribution and *vice versa*.

Specifically, we compute uncertainty of a sample by measuring the variance of model outputs of perturbed samples by various transformation methods for data augmentation: including color jitter, shear, and cutout [12] (illustrated in Figure 3). Following the derivation from Gal *et al*. [14], we approximate the uncertainty by Monte-Carlo (MC) method of the distribution $p(y=c|x)$ when given the prior of the perturbed sample $\tilde{x}$, as $p(\tilde{x}|x)$. We define the perturbation prior $p(\tilde{x}|x)$, as a uniform mixture of the various perturbations as shown in the examples in Figure 3. The derivation can be written as:

$$\begin{aligned} p(y=c|x) &= \int_{\tilde{\mathcal{D}}} p(y=c|\tilde{x}_t)p(\tilde{x}_t|x)d\tilde{x}_t \\ &\approx \frac{1}{A}\sum_{t=1}^{A} p(y=c|\tilde{x}_t), \end{aligned} \tag{1}$$

where x, $\tilde{x}$, y and A denote a sample, a perturbed sample, the label of the sample, and the number of perturbation methods, respectively. The distribution $\tilde{\mathcal{D}}$ denotes the data distribution defined by the perturbed samples $\tilde{x}$. In particular, the perturbed sample $\tilde{x}$ is drawn by a random function $f_r(\cdot)$, as:

$$\tilde{x} = f_r(x|\theta_r), r = 1, ..., R, \tag{2}$$

where θ_r is a hyper-parameter which denotes the random factor of the r-th perturbation. The prior $p(\tilde{x}|x)$ is defined as:

$$\tilde{x} \sim \sum_{r=1}^{R} w_r * f_r(x|\theta_r), \tag{3}$$

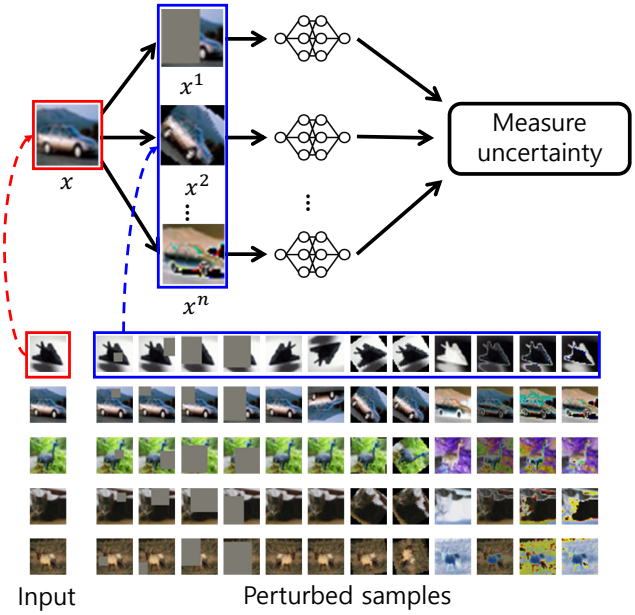


Figure 3: Estimating uncertainty of a data sample (x) with its perturbed samples ($\tilde{x}$) for the proposed Rainbow Memory. Detailed procedure is summarized in Algorithm 1.

where the random variable $w_r, r = \{1, ..., R\}$ is drawn from a categorical binary distribution. From the approximated distribution (1), we measure the uncertainty of the sample with respect to the perturbation as:

$$\begin{aligned} S_c &= \sum_{t=1}^{T} \mathbb{1}_c \underset{\hat{c}}{\operatorname{argmax}}\, p(y=\hat{c}|\tilde{x}_t), \\ u(x) &= 1 - \frac{1}{T}\max_c S_c, \end{aligned} \tag{4}$$

where $u(x)$ denotes the uncertainty of the sample x and S_c is the number of times that class c is the predicted top-1 class. The $\mathbb{1}_c$ denotes the binary class indexing vector. The lower valued $u(x)$, corresponding to more consistent top-1 class over perturbations, indicates that x resides in a region where a model is strongly confident.

Algorithm 1 summarizes our proposed diversity-aware memory update algorithm. Following GDumb [38], we also assign the same amount of memory slots (k_c) over the 'seen' classes (N). After assigning the exemplars to the memory slots, we compute the uncertainties for both streamed samples ($\mathfrak{D}_t^S$) and stored exemplars ($\mathfrak{D}_{t-1}^M$) in a memory at task t, then sort all these samples ($\mathfrak{D}_c$) by their uncertainties. From the sorted list, we select samples with an interval of $|\mathfrak{D}_c|/k_c$ to secure the diversity. As a result of this sampling, we fill the memory with exemplars in a wide spectrum ranged from strongly perturbation, *i.e.*, robust samples, to fragile ones. This imposes perturbation-based diversity to the episodic memory.

Algorithm 1 Diversity-Aware Memory Update

Input: K denotes memory size, N_t denotes the number of seen classes until task t, $\mathfrak{D}_t^S$ denotes stream data at task t, $\mathfrak{D}_{t-1}^M$ denotes exemplars stored in a episodic memory after task $t-1$.
Output: $\mathfrak{D}_t^M$ exemplars after learning task t.
$\mathfrak{D}_t^M = \{\}$ ▷ New exemplars from scratch
$k_c = floor(K/N_t)$ ▷ Class-balanced sampling
for $c = 1, 2, \ldots, N_t$ **do**
  $\mathfrak{D}_c = \{(x, y) | y = c, (x, y) \in \mathfrak{D}_t^S \cup \mathfrak{D}_{t-1}^M\}$
  Sort $\mathfrak{D}_c$ by $u(x)$ computed by (4)
  for $j = 1, 2, \ldots, k_c$ **do**
    $i = j * |\mathfrak{D}_c|/k_c$ ▷ $|\mathfrak{D}_c|/k_c$ step-size indexing
    $\mathfrak{D}_t^M$ += $\mathfrak{D}_c[i]$
  end for
end for

4.2. Diversity Enhancement by Augmentation

To further enhance the diversity of exemplars from the memory, we employ data augmentation (DA). The DA's diversify a given set of samples by image-level or feature-level perturbations, which correspond to the philosophy of updating memory by securing the diversity (Section 4.1).

We consider various perturbation types including simple single-image-based DA perturbing the original input image, mixed-labeled DA which integrates multiple images [49, 51] and automated DAs (AutoDAs) [9, 11, 32]. The stochastically chosen various augmentations succeed in image classification. Yet, the efficacy of the DA's has not been well investigated in the CIL context.

Mixed-Label Data Augmentation. As task iteration proceeds, the samples in a new task are likely to follow different distribution from the one in the episodic memory (*i.e.*, from the previous tasks). We adopt mixed-labeled DA to 'mix' images in the classes of the new tasks and the exemplars of the old classes in the memory. This mixed-label DA alleviates the side effects caused by change of class distribution over the tasks and improves the performances.

As one of the representative mixed-labeled DA methods, CutMix [49] generates a mixed sample and a smoothed label, given the set of supervised samples (x_1, y_1) and (x_2, y_2), as:

$$\begin{aligned} \tilde{x} &= \mathbf{m} \odot x_1 + (\mathbf{1} - \mathbf{m}) \odot x_2, \\ \tilde{y} &= \lambda y_1 + (1 - \lambda) y_2, \end{aligned} \tag{5}$$

where the set $\mathbf{m}$ denotes the randomly selected pixel region for the image x_1 according to the hyper-parameter β drawn from the beta-distribution. As shown in (5), the mixed-label DA generates artificial samples that are hard to be considered as a variant of the source images unlike the conventional data augmentations manipulating an original image by flipping, rotating, and/or contrasting while not ruining a class boundary.

Automated Data Augmentation. In addition to the above mixed-labeled DAs, we further use AutoDA to enrich the augmentation effect by compositing multiple DA's on the model performance under CIL. Especially, we employ AutoAugment [9], providing parameters for determining the number of augmentations and their magnitudes.

5. Experiments

We empirically validate the efficacy of our RM by comparing it with state of the arts in various experimental setups; CIL task setups for the benchmarks, memory-sizes of episodic memory, and performance metrics. In addition, we further investigate components of the propose RM; memory management strategy and augmentation methods for their contribution to the CIL performances.

5.1. Experimental Setup

Benchmark Task Setup. We evaluate algorithms mostly in blurry-CIL setup, otherwise stated. Following [1], we denote blurry-CIL setup as 'Blurry$M$', where the M denotes the portion of samples coming from the other tasks. Therefore, each task in the blurry-CIL setup contains samples from its assigned major classes (*i.e.*, the most frequent classes and assigned to each task exclusively) consisting of $(100 - M)\%$ and ones of minor classes (*i.e.*, the other classes of C except for the assigned major classes) consisting of remaining $M\%$. Note that the class distribution of minor classes in each task are balanced.

In addition, we consider two different learning setup; *online* and *offline*. In online, the incoming samples are presented to a model only once except the ones selected as exemplars since it does not have a buffer which is large enough to keep the whole streamed samples. On the other hand, in offline, a model can observe the incoming samples multiple times (*i.e.*, epochs) with the buffer. Please note that we repeat each experiment three times to report means and standard deviations except the ImageNet experiments.

Datasets and Metrics. We use MNIST, CIFAR10, CIFAR100 and ImageNet (ILSVRC2012) datasets to configure CIL task setups for evaluations. We randomly split and assign with different random seeds a set of all classes (C) into 5 tasks to generate a CIL task setup, and thus each class-subset (T_t) has 2 and 20 major classes for CIFAR10 and CIFAR100 datasets, respectively. For ImageNet, we split 1000 classes to 10 tasks, so each class-subset (T_t) has 100 major classes.

We use three popular metrics in the literature, such as *Last Accuracy (A5)*, *Last Forgetting (F5)*, and *Intransigence (I5)*. 'Last' refers to the value is measured after all tasks

are learned, and we denote it with number '5' here because both of CIFAR10 and CIFAR100 have five tasks. Accordingly, they will be A10, F10, and I10 for ImageNet. Please refer to the supplementary material for more details about the metrics. Finally, we use various episodic memory sizes for different datasets as the size of the datasets differ.

Baselines and Implementation Details. We compare our proposed RM with the standard CIL methods including EWC [27], Rwalk [5], iCaRL [39], BiC [48] and GDumb [38], the only method specifically designed for the blurry setup. Note that GSS [1] is not compared since GDumb outperforms it by large margins. The comparable CIL methods utilize MLP400, ResNet18, ResNet32, and ResNet34 [19] as their network architectures for MNIST, CIFAR10, CIFAR100, and ImageNet, respectively. For CIFAR10/100, we use the same backbone to the official GDumb [38] implementation[1] throughout all experiments. For ImageNet, we use the backbone from their original implementation [19].

For the training hyperparameters of experiments on MNIST and CIFAR10/100, we use batch-size of 16, cosine annealing learning-rate schedule ranged from 0.05 to 0.0005, and the number of epochs of 256, following [38]. For those on ImageNet, we use batch-size of 256, step annealing learning-rate schedule ranged from 0.1 to 0.001, and the number of epochs of 100, which are used from BiC [48].

In addition, we use an episodic memory, which is updated through reservoir sampling which exhibits the best performance (Section 5.3), to the baselines not considering the existence of memory; EWC and Rwalk, for fair comparison. As expected, all memory-attached baselines outperform the corresponding original ones.

5.2. Results

We compare the propose RM to other methods in 'Blurry10-Online' setup on various datasets and summarize the results in Table 1. As shown in the table, RM consistently outperforms all other methods, and the gain becomes larger when the number of classes ($|C|$) increases, which is more challenging. Note that the original BiC performs significantly worse in ImageNet in the blurry setup, so we eliminate the distilling loss yielding irregular values, then BiC performs reasonably well (denoted by * in Table 1). On MNIST, however, RM without DA performs the best. We believe that DA interferes the model training with perturbed samples since the exemplars are enough to avoid forgetting. On the other hand, DA improves the metrics with large margins on the other datasets as we expected in section 4.2.

Table 2 presents the comparison on CIFAR10-Blurry10-Online for three episodic memory sizes (K); 200, 500 and 1,000. We again observe that our proposed RM outperforms all other baselines over all three memory-sizes in terms of A5, F5, and I5 by significant margins in Blurry and online CIL setup on CIFAR10. It is interesting that EWC and Rwalk do not perform well in forgetting (F5) despite their competitive A5 scores regardless of the memory size. The results imply that these methods preserve effective exemplars in the final task, which are enough to restore the forgetting happening in the previous tasks. iCaRL, GDumb and BiC are less effective for intransigence (I5) with larger memory size while they perform well in forgetting compared to EWC and Rwalk as a tradeoff.

Our RM not only outperforms other baselines for accuracy but also exhibit good forgetting and intransigence performance, regardless of memory sizes. It is also observed that the performance gaps between ours and the others decrease when the memory-size becomes larger since the impact of sampling efficiency decreases with redundant samples. Note that these results on CIFAR10 exhibit similar trends to the results on CIFAR100 and ImageNet (shown in Table 1). Although the CIFAR100 and ImageNet has $10\times$ or $100\times$ more classes than the CIFAR10, RM still outperforms all the baselines in all three metrics by large margins. These results imply that our RM is quite effective for more practical and realistic CIL setup of blurry and online, compared to the prior arts.

5.3. Detailed Analysis

On Various Blurry Levels. Even though blurry-CIL is the main task of our interests, it is interesting to investigate the performance of the proposed RM on disjoint-CIL (*i.e.*, Blurry0) setup and in various blurry levels. We summarize the comparative results in Table 3.

In disjoint-CIL where catastrophic forgetting is more severe than blurry-CIL, regularization-based methods such as EWC [27] and Rwalk [5] show competitive performances. It is expected that disjoint-CIL setup tends to exaggerate catastrophic forgetting that regularization-based methods aim to address (Section 3). Notably, RM performs comparably without any regularization while outperforming rehearsal-based methods, *e.g.*, iCaRL, GDumb and BiC.

In the offline setup, the gain by RM diminishes and prior arts slightly outperform the RM. We conjecture that keeping the large incoming samples in buffer dilutes the sensitivity of exemplar sampling. In blurry-CIL setups with online-setting (Blurry10 and Blurry30), RM outperforms other baselines by remarkable margins even when DA is not applied. With the proposed DA, RM achieves over 70% accuracy for both Blurry10 and Blurry30 setups, far better than the other baselines.

We further compare the accuracy trajectories over the task streams; three streams generated from stochastically assigned function, $\psi(c)$, with different random seeds, for CIFAR10 and single stream for ImageNet and summarize

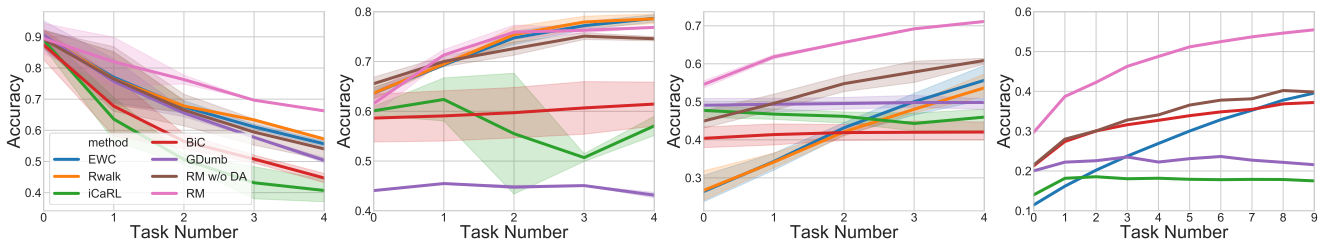
[1] https://github.com/drimpossible/GDumb

Table 1: Comparison with three metrics (A{5, 10}, F{5, 10}, and I{5, 10}: %) in {MNIST, CIFAR100, ImageNet}-Blurry10-Online. * indicates the reproduction of BiC with only using classification loss without distilling loss to be better suited for Blurry10 setup.

Methods	MNIST (K=500)			CIFAR100 (K=2,000)			ImageNet (K=20,000)		
	A5 (↑)	F5 (↓)	I5 (↓)	A5 (↑)	F5 (↓)	I5 (↓)	A10 (↑)	F10 (↓)	I10 (↓)
EWC	90.98 ± 0.61	4.23 ± 0.45	4.54 ± 0.94	26.95 ± 0.36	11.47 ± 1.26	43.18 ± 14.22	39.54	14.41	42.68
Rwalk	90.69 ± 0.62	4.77 ± 0.36	4.96 ± 0.56	32.31 ± 0.78	15.57 ± 2.17	37.18 ± 10.02	35.26	13.92	46.96
iCaRL	78.09 ± 0.60	6.09 ± 0.23	17.03 ± 0.60	17.39 ± 1.04	5.38 ± 0.88	44.18 ± 9.29	17.52	1.94	81.94
GDumb	88.51 ± 0.52	2.67 ± 0.31	6.75 ± 0.43	27.19 ± 0.65	7.49 ± 0.95	41.18 ± 7.23	21.52	4.07	60.70
BiC	77.75 ± 1.27	8.25 ± 1.45	17.37 ± 1.27	13.01 ± 0.24	4.63 ± 0.46	53.84 ± 11.85	37.20*	1.52*	45.02*
RM w/o DA	**92.65 ± 0.33**	**0.58 ± 0.09**	**3.14 ± 0.94**	34.09 ± 1.41	**4.01 ± 0.50**	34.51 ± 4.58	37.96	2.63	44.26
RM	91.80 ± 0.69	0.75 ± 0.30	3.62 ± 0.63	**41.35 ± 0.95**	4.99 ± 0.89	**20.18 ± 3.06**	**50.11**	**1.39**	**32.11**

Table 2: Comparison with three metrics (A5, F5, and I5: %) for three episodic memory sizes in CIFAR10-Blurry10-Online. DA is used in RM denotes CutMix+AutoAug.

Methods	K=200			K=500			K=1,000		
	A5 (↑)	F5 (↓)	I5 (↓)	A5 (↑)	F5 (↓)	I5 (↓)	A5 (↑)	F5 (↓)	I5 (↓)
EWC	40.07 ± 2.14	21.20 ± 0.76	61.91 ± 4.51	55.65 ± 4.60	16.06 ± 3.89	44.24 ± 11.98	68.67 ± 0.95	12.63 ± 1.78	25.97 ± 10.88
Rwalk	38.66 ± 1.52	20.67 ± 2.36	65.81 ± 4.85	53.66 ± 3.18	17.04 ± 0.31	45.81 ± 9.78	68.20 ± 1.86	11.48 ± 1.19	25.17 ± 11.57
iCaRL	37.43 ± 1.31	2.08 ± 2.23	63.51 ± 13.73	45.98 ± 3.04	4.75 ± 1.70	51.91 ± 2.57	53.60 ± 2.82	7.21 ± 2.58	37.84 ± 13.49
GDumb	35.85 ± 1.03	1.67 ± 3.49	55.31 ± 6.02	49.47 ± 1.08	1.44 ± 2.77	40.91 ± 14.04	64.26 ± 1.21	0.37 ± 1.92	31.81 ± 13.37
BiC	33.29 ± 0.86	3.91 ± 1.64	50.37 ± 6.96	42.06 ± 2.41	1.34 ± 2.27	52.04 ± 15.50	47.81 ± 3.04	3.03 ± 1.44	52.77 ± 15.54
RM w/o DA	44.41 ± 1.40	0.90 ± 0.93	49.51 ± 11.09	60.87 ± 0.88	0.95 ± 1.14	35.74 ± 13.89	70.93 ± 1.57	**-1.43 ± 0.71**	22.07 ± 14.07
RM	**54.61 ± 1.62**	**-2.60 ± 1.91**	**43.57 ± 11.63**	**71.13 ± 0.25**	**-0.85 ± 0.28**	**18.29 ± 14.21**	**78.04 ± 0.50**	1.29 ± 1.26	**11.64 ± 5.83**



(a) CIFAR10-Disjoint-Online (b) CIFAR10-Blurry10-Offline (c) CIFAR10-Blurry10-Online (d) ImageNet-Blurry10-Online

Figure 4: Illustration of accuracy changes as tasks are being learned in (a) CIFAR10-Disjoint-Online, (b) CIFAR10-Blurry10-Offline, (c) CIFAR10-Blurry10-Online, (d) ImageNet-Blurry10-Online settings. More results are presented in the supplement.

the results in Figure 4. For the online settings ((a), (c) and (d)), our RM consistently outperforms the other baselines over entire task stream. However, in offline setting (b), RM comparably performs to the prior arts over the entire task stream as summarized in Table 3.

Uncertainty Measure. We compare three methods for estimating uncertainty by various Monte-Calro methods; (1) no MC (No_MC), (2) RandAug-based (RandAug_MC), and (3) AutoAug [9]-based methods (AutoAug_MC), summarizing the A5 results in Table 5. Note that RandAug_MC and AutoAug_MC also leverage configuring the stochastic data perturbation set as well as DA during training.

As shown in the table, the two automated DA-based methods improve the accuracy compared to the No_MC case, caused by diversity-enhanced configuration. For measuring the uncertainty in our RM, we use RandAug_MC.

Comparison to Other Memory Update Algorithms. To investigate exclusive gains by the memory update algorithm, we compare RM with other memory update schemes while leaving other components unchanged and summarize results in Table 6. The other algorithms include *Random*, *Reservoir* [44] and *Prototype* [39]. Random selects new exemplars for the next episodic memory randomly from current exemplars and incoming samples. Reservoir conducts uniform random sampling on a unknown length task stream. The prototype selects the samples where the extracted features are close to the feature mean of its own class. As shown in the table, RM outperforms all the augmentation conditions with different settings of K.

Data Augmentation. We also investigate the effects of various DA methods on performances by comparing the adopted DA methods with others while other components

Table 3: Comparison of last accuracy (A5 (↑), %) over benchmarks {Disjoint (0%), Blurry (10%), and Blurry (30%)} and training setups {Online and Offline} on CIFAR10 (K=500).

Methods	Blurry0 (=Disjoint)		Blurry10		Blurry30	
	Online	Offline	Online	Offline	Online	Offline
EWC	55.66 ± 1.18	64.00 ± 1.34	55.65 ± 4.60	**78.67 ± 1.06**	60.57 ± 1.15	85.00 ±0.42
Rwalk	55.91 ± 1.85	65.04 ± 0.11	53.66 ± 3.18	78.59 ± 1.37	59.03 ± 0.05	**85.18 ± 0.57**
iCaRL	40.70 ± 5.13	**65.61 ± 2.57**	45.98 ± 3.04	57.07 ± 2.74	48.11 ± 4.63	64.90 ± 7.95
GDumb	50.37 ± 1.17	42.47 ± 1.15	46.70 ± 1.53	43.16 ± 0.77	47.78 ± 3.77	45.72 ± 0.64
BiC	44.70 ± 2.12	59.53 ± 4.30	42.06 ± 2.41	61.45 ± 6.25	42.92 ± 1.47	71.93 ± 2.45
RM w/o DA	54.05 ± 4.94	59.47 ± 0.61	60.87 ± 0.88	74.58 ± 0.60	60.92 ± 6.48	83.91 ± 0.40
RM	**66.25 ± 0.21**	61.91 ± 0.63	**71.13 ± 0.18**	76.86 ± 0.04	**73.90 ± 0.80**	85.10 ± 0.16

Table 4: Comparison of last accuracy (A5 (↑), %) over methods with data augmentations in CIFAR10-Blurry10-Online. The results on $K = 1,000$ is reported in the supplementary material. 'CM+AA' refers to CutMix+AutoAug.

Methods	K=200					K=500				
	None	CutMix	RandAug	AutoAug	CM+AA	None	CutMix	RandAug	AutoAug	CM+AA
EWC	40.0±2.1	41.9±1.0	44.7±0.6	48.3±3.5	50.3±1.2	55.6±4.6	56.2±0.7	60.0±5.3	64.8±0.6	67.5±0.9
Rwalk	38.6±1.5	41.3±2.2	46.5±2.9	48.7±2.7	51.8±1.6	53.6±3.1	57.5±1.4	62.5±3.0	64.7±1.0	67.2±1.5
iCaRL	37.4±1.3	37.9±3.8	38.4±1.4	41.8±2.3	43.3±2.2	45.9±3.0	46.9±1.4	51.3±1.1	51.6±2.8	56.6±1.2
GDumb	33.3±2.0	35.8±1.0	37.1±2.0	38.4±1.1	41.4±1.1	46.7±1.5	49.4±1.0	54.3±1.4	55.9±1.4	58.2±2.7
BiC	33.2±0.8	33.2±0.8	27.1±2.7	29.7±3.1	31.2±0.7	42.0±2.4	42.0±2.4	38.6±2.8	38.7±1.5	38.4±2.5
RM	**44.4±1.4**	**45.9±2.4**	**49.9±2.9**	**55.3±2.2**	**54.6±1.6**	**60.8±0.8**	**62.0±3.5**	**68.6±0.7**	**69.6±2.9**	**71.1±0.1**

Table 5: Comparison of uncertainty measures for RM on CIFAR10-Blurry10-Online (K=500).

	No_MC	RandAug_MC	AutoAug_MC
A5 (%)	58.59	61.27	60.1

Table 6: Comparison of last accuracy (A5 (↑), %) over memory update methods without data augmentations in CIFAR10-Blurry10-Online. 'CM+AA' refers to CutMix+AutoAug.

Methods	K=200			K=1,000		
	None	CutMix	CM+AA	None	CutMix	CM+AA
Random	24.1 ± 1.4	24.0 ± 1.0	22.4 ± 0.8	46.7 ± 2.5	52.5 ± 4.2	52.7 ± 2.8
Reservoir	38.0 ± 2.2	39.1 ± 0.8	49.4 ± 1.8	64.6 ± 4.2	67.2 ± 5.3	75.5 ± 0.0
Prototype	34.6 ± 0.5	33.8 ± 1.9	26.5 ± 3.9	48.1 ± 5.7	41.1 ± 4.1	29.3 ± 1.5
Uncertainty (RM)	**43.8 ± 1.2**	**42.4 ± 1.8**	**52.2 ± 1.3**	**64.7 ± 4.1**	**71.8 ± 4.3**	**76.1 ± 1.1**

unchanged in Table 4. As shown in the table, other methods also enjoyed the performance enhancement by DA same as RM. However, the enhancement from CutMix + AutoAug used for RM is the most effective among all DAs. Note that even when adding various DA, RM achieves the best performance surpassing the other baselines.

6. Conclusion

We address a realistic and real-world class incremental (continual) learning setup where tasks share the classes, denoted as blurry-CIL. To effectively address such scenario, we propose to enhance diversity of samples in an episodic (or representative) memory. Specifically, we propose a new diversity-enhanced sampling method using per-sample perturbation-based uncertainty. In addition, we employ diverse sets of data augmentation techniques to further improve the diversity, that is representativeness and discriminativeness of exemplars, induced from the proposed memory update.

In blurry-CIL scenarios on CIFAR10, CIFAR100, and ImageNet, our diversity-enhancing method (named Rainbow Memory or RM) not only outperforms the state-of-the-art methods by large margins but also presents comparable performances on disjoint and offline CIL setups. We further investigate the effectiveness of the proposed method in various blurry setups and even in the disjoint setup, along with in-depth analysis for each proposed components. As a future work, we will investigate the relationships between uncertainty-based memory update and data augmentation in training time and their effects on diverse CIL tasks.

Acknowledgement. JC is partly supported by the National Research Foundation of Korea (NRF) (No.2019R1C1C1009283) and Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-01842, Artificial Intelligence Graduate School Program (GIST) and No.2019-0-01351, Development of Ultra Low-Power Mobile Deep Learning Semiconductor With Compression/Decompression of Activation/Kernel Data), and Center for Applied Research in Artificial Intelligence (CARAI) grant funded by Defense Acquisition Program Administration (DAPA) and Agency for Defense Development (ADD) (UD190031RD). All authors thank Sungmin Cha (NAVER AI Lab) and Hyeonseo Koh (GIST) for discussions, and NAVER Smart Machine Learning (NSML) [25] team for GPU support.

References

[1] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *NeurIPS*, pages 11816–11825, 2019. 2, 3, 5, 6

[2] Yogesh Balaji, Mehrdad Farajtabar, Dong Yin, Alex Mott, and Ang Li. The effectiveness of memory replay in large scale continual learning. *arXiv preprint arXiv:2010.02418*, 2020. 3

[3] Zalán Borsos, Mojmír Mutný, and A. Krause. Coresets via bilevel optimization for continual learning and streaming. In *NeurIPS*, 2020. 3

[4] Francisco M. Castro, Manuel J. Marin-Jimenez, Nicolas Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *ECCV*, 2018. 2, 3

[5] Arslan Chaudhry, Puneet K. Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *ECCV*, 2018. 1, 2, 3, 6

[6] Arslan Chaudhry, Puneet K. Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *ECCV*, 2018. 2, 3

[7] Arslan Chaudhry, Albert Gordo, Puneet K Dokania, Philip Torr, and David Lopez-Paz. Using hindsight to anchor past knowledge in continual learning. *arXiv preprint arXiv:2002.08165*, 2020. 3

[8] Yulai Cong, Miaoyun Zhao, J. Li, Sijia Wang, and L. Carin. GAN memory with no forgetting. In *NeurIPS*, 2020. 3

[9] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. AutoAugment: Learning augmentation strategies from data. In *CVPR*, June 2019. 2, 5, 7

[10] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical data augmentation with no separate search. *arXiv preprint arXiv:1909.13719*, 2(4):7, 2019. 2

[11] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR Workshops*, pages 702–703, 2020. 5

[12] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 4

[13] Enrico Fini, Stéphane Lathuilière, E. Sangineto, Moin Nabi, and E. Ricci. Online continual learning under extreme memory constraints. In *ECCV*, 2020. 2, 3

[14] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, pages 1050–1059, 2016. 4

[15] Alexander Gepperth and Barbara Hammer. Incremental learning algorithms and applications. In *European Symposium on Artificial Neural Networks (ESANN)*, Bruges, Belgium, 2016. 2

[16] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013. 1

[17] Yunhui Guo, Mingrui Liu, Tianbao Yang, and Tajana Rosing. Improved schemes for episodic memory-based lifelong learning. *Advances in Neural Information Processing Systems*, 33, 2020. 3

[18] T. Hayes, Kushal Kafle, Robik Shrestha, Manoj Acharya, and Christopher Kanan. Remind your neural network to prevent catastrophic forgetting. In *ECCV*, 2020. 3

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6

[20] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. *arXiv preprint arXiv:1902.00751*, 2019. 1

[21] Wenpeng Hu, Zhou Lin, Bing Liu, Chongyang Tao, Zhengwei Tao, Jinwen Ma, Dongyan Zhao, and Rui Yan. Overcoming catastrophic forgetting via model adaptation. In *ICLR*, 2019. 3

[22] Ahmet Iscen, J. Zhang, S. Lazebnik, and C. Schmid. Memory-efficient incremental learning through feature adaptation. In *ECCV*, 2020. 3

[23] Xisen Jin, Junyi Du, and Xiang Ren. Gradient based memory editing for task-free continual learning. *arXiv preprint arXiv:2006.15294*, 2020. 2

[24] Woo-Young Kang and Byoung-Tak Zhang. Continual learning with generative replay via discriminative variational autoencoder. In *NeurIPS Workshop on Continual Learning*, 2018. 1

[25] Hanjoo Kim, Minkyu Kim, Dongjoo Seo, Jinwoong Kim, Heungseok Park, Soeun Park, Hyunwoo Jo, KyungHyun Kim, Youngil Yang, Youngkwan Kim, et al. Nsml: Meet the mlaas platform with a real-world case study. *arXiv preprint arXiv:1810.09957*, 2018. 8

[26] Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. *arXiv preprint arXiv:2009.06962*, 2020. 2

[27] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 1, 6

[28] James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *PNAS*, 2017. 2

[29] Sang-Woo Lee, Jin-Hwa Kim, JungWoo Ha, and Byoung-Tak Zhang. Overcoming catastrophic forgetting by incremental moment matching. In *NeurIPS*, 2017. 1, 2

[30] Yuanpeng Li, Liang Zhao, Kenneth Church, and Mohamed Elhoseiny. Compositional language continual learning. In *ICLR*, 2019. 1

[31] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Trans. on PAMI*, 2017. 2

[32] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. In *NeurIPS*, pages 6665–6675, 2019. 2, 5

[33] Xialei Liu, Marc Masana, Luis Herranz, Joost van de Weijer, Antonio M. López, and Andrew D. Bagdanov. Rotate your networks: Better weight consolidation and less catastrophic forgetting. In *ICPR*, 2018. 2

[34] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12245–12254, 2020. 3

[35] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *NeurIPS*, 2017. 3

[36] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *ECCV*, 2018. 2

[37] German Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 2018. 1, 3

[38] Ameya Prabhu, P. Torr, and Puneet K. Dokania. GDumb: A simple approach that questions our progress in continual learning. In *ECCV*, 2020. 1, 2, 3, 4, 6

[39] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. In *CVPR*, 2017. 1, 2, 3, 6, 7

[40] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, , and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *ICLR*, 2019. 3

[41] Ari Seff, Alex Beatson, Daniel Suo, and Han Liu. Continual learning in generative adversarial nets. *CoRR*, abs/1705.08395, 2017. 3

[42] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *NeurIPS*, 2017. 1, 3

[43] Gido M. van de Ven and Andreas S. Tolias. Three continual learning scenarios and a case for generative replay. In *NeurIPS Workshop on Continual Learning*, 2018. 1, 2

[44] Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57, 1985. 7

[45] Max Welling. Herding dynamical weights to learn. In *ICML*, 2009. 3

[46] Chenshen Wu, Luis Herranz, Xialei Liu, Yaxing Wang, Joost van de Weijer, and Bogdan Raducanu. Memory Replay GANs: learning to generate images from new categories without forgetting. In *NeurIPS*, 2018. 1, 3

[47] Sen Wu, Hongyang R Zhang, Gregory Valiant, and Christopher Ré. On the generalization effects of linear transformations in data augmentation. *arXiv preprint arXiv:2005.00695*, 2020. 2

[48] Yue Wu, Yan-Jia Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *CVPR*, 2019. 2, 3, 6

[49] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, pages 6023–6032, 2019. 2, 5

[50] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, 2017. 2

[51] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 5