Is Continual Learning Truly Learning Representations Continually?

Sungmin Cha¹,2, Dongsub Shim², Hyunwoo Kim², Moontae Lee²,³, Honglak Lee²,⁴, and Taesup Moon¹∗

¹ Seoul National University ² LG AI Research ³ University of Illinois at Chicago ⁴ University of Michigan
sungmin.cha@snu.ac.kr, {dongsup.shim, hyunwoo.kim}@lgai.research, {moontae.lee, honglak.lee}@lgai.research, tsmoon@snu.ac.kr

Abstract

Continual learning (CL) aims to learn from sequentially arriving tasks without forgetting previous tasks. Whereas CL algorithms have tried to achieve higher average test accuracy across all the tasks learned so far, learning continuously useful representations is critical for successful generalization and downstream transfer. To measure representational quality, we re-train only the output layers using a small balanced dataset for all the tasks, evaluating the average accuracy without any biased predictions toward the current task. We also test on several downstream tasks, measuring transfer learning accuracy of the learned representations. By testing our new formalism on ImageNet-100 and ImageNet-1000, we find that using more exemplar memory is the only option to make a meaningful difference in learned representations, and most of the regularization- or distillation-based CL algorithms that use the exemplar memory fail to learn continuously useful representations in class-incremental learning. Surprisingly, unsupervised (or self-supervised) CL with sufficient memory size can achieve comparable performance to the supervised counterparts. Considering non-trivial labeling costs, we claim that finding more efficient unsupervised CL algorithms that minimally use exemplary memory would be the next promising direction for CL research.

1 Introduction

Continual learning (CL), which aims to efficiently learn sequentially arriving tasks without forgetting past tasks, has been considered as a holy grail in machine learning research. Recently, a plethora of neural network based CL algorithms have been proposed [34, 14, 32], particularly in the continual classification setting. Namely, when a training dataset for novel classes arrive, those algorithms aim to update the classifier that can well predict both the new and the past learned classes. To that end, the effectiveness of those algorithms were typically evaluated with the average accuracy across all the classes learned so far, since it is regarded as a good proxy for measuring both the plasticity (for learning new classes) and the stability (for not forgetting past classes).

In this paper, we ask whether above evaluation protocol for CL algorithms is truly measuring what those algorithms are learning. Namely, in class-incremental learning (Class-IL), for example, since the task boundaries are not given at the test time, a critical problem is to resolve the so-called biased prediction toward the current task, and many recent work focused on mitigating such bias by rectifying or post-processing the output classification layer [44, 22, 2, 5, 6]. However, it is not clear whether the average accuracy gain we get via such processing truly reflect the gains the algorithms...
achieve via continually learning the representations of the tasks learned so far. Furthermore, in many recent vision applications, the classification models are used as a backbone model to transfer the general representations learned from a classification task to diverse downstream tasks, such as object detection or semantic segmentation. To that end, it is again not clear whether a continually learned classifier that is favorable with respect to the average accuracy can actually show gradually improving downstream task performance as the continual learning step increases.

To address above question, we propose the re-consideration of the purpose of CL in classification. We re-define the concept of continual representation learning, which aims to accumulate representations during CL, and propose two new evaluation criteria as surrogates for evaluating the learned representations of CL algorithms. Namely, after each continual learning step, (a) re-train the output layers using a small balanced dataset for all the tasks and evaluate the average test accuracy so that the effect of the biased predictions toward the current task can be removed, and (b) test on several downstream tasks to obtain the transfer learning accuracy. Using these criteria, we evaluate several popular CL algorithms on both ImageNet-100 and ImageNet-1000 datasets and obtain following findings: First, the biased prediction in class-incremental learning (Class-IL) exaggerates the actual forgetting in the learned representations. Second, the biased prediction can be easily removed by the proposed technique of retraining the output layer using a small balanced dataset. Third, most of the latest regularization (or distillation)-based CL algorithms using the exemplar memory fail to accumulate representations continually, especially in Class-IL, whereas they work relatively well in data-incremental learning (Data-IL). Additionally, only using more exemplar memory clearly helps to increase the quality of learned representations. Finally, the unsupervised (or self-supervised) CL with enough size of exemplar memory can achieve performances comparable to those of supervised CL methods, with only requiring minimum labeling cost.

2 Related Work

Supervised continual learning The type of CL methods is categorized into three different types [14]. First, regularization-based methods overcome the catastrophic forgetting by maintaining important weights for previous tasks at the training time of the current task. For measuring important weights, several papers have suggested different methods, showing superior performance, especially for task-incremental learning, but degraded performance for class-incremental learning [27, 39, 1, 25, 33, 8]. Also, distillation-based methods can be considered as one subpart of it and focus on devising a distillation method which overcome the catastrophic forgetting problem [29, 17, 7]. Second, dynamic architecture-based approaches dynamically extend the capacity of neural networks when it is required to learn a new task without the catastrophic forgetting [38, 45, 31, 39, 24, 28]. However, the weakness of those methods is known as the complexity of the method, applicability to a large-scale dataset, and requiring somewhat more hyperparameters. Finally, exemplar-based methods were considered the most promising approach, showing superior performance in most CL scenarios [37, 6, 44, 22, 35, 2]. To overcome the catastrophic forgetting, it maintains a tiny exemplar memory saving a subset of previous task’s dataset and use them when the model is trained for a new task [10]. However, because of the imbalance between the current and exemplar data in a mini-batch, the model encounters the biased prediction problem and many methods are devised to alleviate it [44, 22, 2, 5]. In general, the exemplar-based method is combined with the regularization (or distillation)-based method together, achieving the most powerful performance in various datasets including the large-scale dataset (e.g., ImageNet) [44, 22, 17].

Unsupervised continual learning Recently, CL using unsupervised (self-supervised) learning has started to discuss [35, 30, 23]. Those papers show the possibility of CL using the unsupervised dataset but the perspective is slightly different each other. [35] first adopts the concept of unsupervised continual learning and proposes a novel approach to learning class-discriminative representations without any knowledge about task identity. Based on research for the self-supervised learning which has grown rapidly recently, [30] first shows that the unsupervised CL can surpass the result of the supervised CL algorithms in the task-incremental learning scenario using small scale datasets (e.g., CIFAR-10/100 and Tiny-ImageNet). Also, they showed that the unsupervised CL less suffers from catastrophic forgetting because it achieves a wider local minima. Another study [23] more focuses on the advantage of the unsupervised CL in a large scale dataset (e.g., ImageNet-1000) in terms of getting a pre-trained model with CL. Without consideration for the traditional CL scenarios (e.g., Class-, Domain- and Class-Incremental Learning [40]), they demonstrated that a competitive pre-trained
model can be obtained through a stream of the unsupervised dataset, by evaluating it with various downstream tasks.

**Self-supervised contrastive learning** Due to the expensive labeling cost for gathering labeled images, learning visual representations from the unsupervised dataset has been actively conducting a lot of research (e.g., pretext task-based [3][16], autoencoder-based [41][47] and contrastive loss-based approach [19]). Among them, contrastive learning has become the major approach, learning representations by pulling representations of positive images but pushing representations of negative images each other [19][45]. In a short period of time, various approaches have been proposed for efficient unsupervised learning with the contrastive loss [24][11][46]. However, it is generally known that a powerful representation can be learned when a large amount of negative samples are available at the training time.

### 3 Problem Formulation and Preliminaries

#### 3.1 Problem formulation and three scenarios of continual learning

Given a sequence of tasks, let $t \in \{1, \ldots, T\}$ represent the $t^{th}$ task. Each task-specific dataset $D_t = \{(x_i^{(b)}, y_i^{(b)})\}_{i=1}^N$ consists of $N$ pairs of an input image and its target label. We assume that $y_i$ is sampled from a task-specific class set $C_t$ such that $y_i \in C_t$. Note that $y_i$ is only available for supervised continual learning. In the case of unsupervised (or self-supervised) continual learning, only the input image $x_i$ is accessible. We train a classification model $f_{\theta} = (g_\phi \circ f_\theta)$, where $f_\theta$ and $g_\phi$ denote an encoder and output layer of the model, on all the tasks consecutively. More specifically, at task $t$, $f_{\theta}$ is trained on $D_t$ for multiple epochs (offline training) without having access to any other datasets, and then $t$ increments by one. An exemplar memory $M_e$ is often employed to store and replay a finite number of data instances from previously seen tasks. By leveraging the target labels, we can balance the class distribution of the samples stored in $M_e$, and the class-balanced samples can be interleaved with $D_t$ for training the classification model $f_{\theta}$. For the cross-entropy (CE) objective function, both the encoder and output layer are trained together. On the other hand, the contrastive learning-based approaches train the encoder and output layer successively or perform classification via the nearest-mean-classifier method.

We consider three general scenarios of continual learning: task-incremental learning (Task-IL), class-incremental learning (Class-IL) and data-incremental learning (Data-IL) by following [40].

**Task-/Class-IL** In these scenarios, each class set $C_t$ has disjoint class labels: $C_j \cap C_k = \emptyset$, $\forall j, k \in \{1, \ldots, T\}$ and $j \neq k$. At inference time, Task-IL provides an additional supervisory signal that indicates the task identity of an input image. The task-ID makes it straightforward to select a dedicated classification layer or head for each task learned during training. The resulting multi-head configuration exhibits less interference between different tasks. Class-IL requires no such extra supervisory signals during inference because it adopts a shared output layer. The single-head configuration, however, compels Class-IL to be more prone to catastrophic forgetting than Task-IL.

**Data-IL** As a part of domain-incremental learning proposed in [40], we propose Data-IL. In this scenario, we consider that each class set $C_t$ contains all classes to be learned during CL: $C_j = C_k$, $\forall j, k \in \{1, \ldots, T\}$ and $C$ denotes a set containing all classes. Each $D_t$ contains pairs of an input image and its target label in $C$ (supervised CL) or only input images (unsupervised CL). In other words, the entire dataset $D$ is evenly distributed for each task such that Data-IL manifests a relatively similar data distribution for all tasks compared to Task- and Class-IL scenarios. Considering Task-/Class-IL and Data-IL at the same time, we can grasp the tendency of each CL algorithm according to the distribution similarity between tasks. Additionally, we believe that Data-IL is more common in the real world, modeling the gradual build-up of new training datasets for a model currently in service.

#### 3.2 GDumb: Greedy Sampler and Dumb Learner

GDumb [35] proposes a simple but practical method that can be generally applied regardless of CL settings or scenarios, such as online/offline CL and Task-/Class–IL. GDumb consists of a Greedy sampler and a Dumb learner. When new training data for the current task arrives during CL, the Greedy sampler updates an exemplar memory using a class-balanced sampling strategy. At inference
time, the Dumb learner (a classification model) is newly trained by the exemplar memory for multiple epochs. Despite this simple idea, GDumb experimentally shows that it outperforms the existing CL algorithms in most settings, showing the limited applicability of those methods in various CL settings.

4 Evaluation from the Perspective of Continual Representation Learning

In this section, we will introduce the concept of continual representation learning and two evaluation techniques. And then, we will report experimental results using the proposed evaluation techniques in the supervised and unsupervised (self-supervised) CL scenarios.

4.1 Motivation: Continual Representation Learning

During the progress of continual learning research in classification, test accuracy-based evaluation metrics, including both forgetting and intransigence measures [9], have been considered as the major metric. In this regard, we would like to raise the question: achieving high test accuracy can be a sufficient evaluation criterion for evaluating CL algorithms? We believe that the ultimate goal of CL is regarded as to get a jointly trained model trained by the whole dataset, in the scenarios of CL. As we already know that, the jointly trained model (e.g., ImageNet-1000 [15] pre-trained model) is not only used for classification, but also is applied to various downstream tasks for transfer learning. Generally, the model that has learned a better representation shows superior performance in both cases at the same time [20]. Therefore, we believe that the model’s encoder trained by CL should be evaluated in more diverse ways, for the proper evaluation of the CL algorithm.

Figure 1: Illustration of continual representation learning. At training time of each task $t$, a model is trained with the current task’s dataset $D_t$ (and the exemplar memory $\mathcal{M}_e$ in the case of exemplar-based CL, see white color box). After training task $t$, the learned representation of the encoder is evaluated by both re-training output layer by using GDumb-based exemplar memory $\mathcal{M}_o$ (red color box) and downstream tasks (blue and yellow color box).

From this point of view, we re-define the concept of continual representation learning (CRL), as shown in Figure 1. First, an encoder of the model sequentially trained in CL should well accumulate representations learned from each task. Second, after training task $t$, the model should not only achieve superior performance in the current CL scenario but also be able to be adapted to various downstream tasks. For evaluating existing CL algorithms in terms of CRL, we propose two evaluation techniques.

Re-training output layer with GDumb-based exemplar memory  At the end of each training task $t$, we maintain the additional exemplar memory $\mathcal{M}_o$ by sampling pairs of $(x_t, y_t)$ in the class-balanced way, and use them to re-train the output layer only. In the supervised CL, exemplars in $\mathcal{M}_o$...
are sequentially sampled from $D_t$, and note that both $\mathcal{M}_o$ and $\mathcal{M}_e$ can share the same exemplars. In the case of the unsupervised CL, each $D_t$ consists of input images $(x_i)$ only but we consider that $|\mathcal{M}_o|$ numbers of a supervised pair $(x_i, y_i)$ are accessible for training the output layer. Note that, by maintaining $\mathcal{M}_o$ and re-training the output layer using it, we can evaluate the learned representations of the encoder trained by each CL algorithm under the same condition, regardless of the supervised or unsupervised CL. Our approach is motivated by GDumb but there are two differences: 1) our $\mathcal{M}_o$ is only used for training the output layer of the model (not the whole model), 2) our proposed method focuses on how to evaluate the learned representations by CL, in a more appropriate manner. From experimental results will be shown in the next section, we confirmed that this approach simply removes the issue of biased predictions in Class-IL.

Evaluation with downstream tasks To evaluate learned representations of the encoder in more diverse ways, we select four downstream tasks including two classification tasks (STL-10 [13] and CUB200 [42] dataset) and one semantic segmentation task (VOC 2012 dataset [18]). In case of experiments with ImageNet-100, we evaluate each encoder with two classification tasks by training the whole model including the encoder. For ImageNet-1000 experiments, we additionally evaluate the representations with the segmentation task by training a PSPNet [48] initialized by the encoder. Note that, by evaluating with downstream tasks, we can check how well the generalizable representations are accumulated during the CL. Note that the ideal result for successful continual representation learning is to achieve superior performance in both the CL scenario and downstream tasks.

4.2 Experimental Analysis of Supervised Continual Learning in ImageNet-100

Experimental settings All experiments are conducted based on the benchmark code proposed by [32], and we used the ResNet-18 model [21] for all experiments. As a state-of-the-art method, we selected SS-IL [2], which reports the result that outperforms PODNet [17], and Instance-wise relation distillation (IRD) of Co2L [7]. We implemented them by referring their official code. Together with three CL scenarios, we consider two CL sequences, such as {10 classes $\times$ 10 tasks} (denoted as 10-Tasks) and {50 classes, 10 classes $\times$ 5 tasks} (denoted as 6-Tasks). When the exemplar memory $\mathcal{M}_e$ is applied for training a model, we marked it as $|\mathcal{M}_e|$ with its size. For GDumb-based exemplar memory $\mathcal{M}_o$ used for re-training the output layer, we sampled 20 pairs of an input image and its target label per each class, marking results of it as GD. Note that, in the case of CE loss-based algorithms, both an encoder and output layer of the model is trained together but supervised contrastive learning(SupCon) [26]-based algorithms are not. Therefore, we reported both the original result and the result after re-training the output layer, for CE loss-based algorithms. The more details on hyperparameters and experimental settings are proposed in Supplementary Materials (S.M).

![Figure 2: Experimental results of Class-IL in the supervised continual learning using ImageNet-100. CE and SupCon denote the used loss function, respectively. GD denotes the result after re-training the output layer with the GDumb-based exemplar memory $\mathcal{M}_o$.](image)

Class-IL. Figure 2(a) reports the experimental results on 10-Tasks with CE loss-based CL algorithms. Joint shows an upper bound result which is trained by datasets until task $t$ and GDumb is the reproduce of [35]. From the figure, we can have following findings: First, naive finetuning (FT) and the regularization-based algorithm (MAS) [4] achieve a worse result than the current state-of-the-art algorithm, SS-IL, in both with and without exemplar memory $\mathcal{M}_e$ cases. The problem, that those algorithms are suffering from, is known as biased prediction. Second, however, the performance of those algorithms (except for SS-IL) significantly increases after re-training the output layer with
GDumb-based exemplar memory (see Alg. (GD)), even surpassing the result of SS-IL. Note that the performance gap between with and without re-training the output layer can be considered as the level of the biased prediction. This experimental results imply that the biased prediction exaggerates the actual forgetting in the learned representations. Also, we observe that the biased prediction is easily solved by the proposed re-training method using $\mathcal{M}_o$. Figure 2(b) shows an experimental result of SupCon-based CL methods. Both MAS and IRD make a difference of learned representations when the exemplar memory $\mathcal{M}_e$ is not applied. However, if the exemplar memory is used for training the model, the effect of both algorithms becomes negligible. Experimental results for three downstream tasks using the trained encoder by each algorithm are shown in Figure 2(c). We observe that the result of all methods incrementally increases. Among them, Joint shows the best result as an upper bound. Similar with the tendency checked in the experiment of Class-IL, using MAS and the exemplar memory $\mathcal{M}_e$ also make an improvement in downstream accuracy. However, in the case of using the exemplar memory, the effect of MAS is minimized. Note that, different from the result in Class-IL, SS-IL achieves a lower performance than all other methods at $t > 1$. On the other hand, we confirm that the performance of both FT(SupCon) and IRD(SupCon) is gradually lowered at each $t$, and the results are proposed in S.M.

We conducted more experiments for other Class-IL algorithms (including LWF [29], LUCIR [22], EEIL [6], and others), including both the scenario of Task-IL and of 6-Tasks. All results are proposed in S.M.

**Data-IL.** By following the definition of Data-IL proposed in the Section 3.1, we conducted experiments with the ImageNet-100 dataset and results are proposed in Figure 3. At test time of each task $t$, we equally evaluate a model with the whole test dataset of ImageNet-100. Note that, the most of Class-IL methods cannot be naturally applied to Data-IL, such as SS-IL and LUCIR [22]. Figure 3(a) and 3(b) show the experimental results of the scenario of 10-Tasks. Because of the high similarity of distribution between tasks, we observe that a natural transfer learning effect enables successful CL even without the exemplar memory $\mathcal{M}_e$ or using other algorithms (see the result of FT). Also, except for the case of MAS(SupCon), using the additional algorithm helps to increase the final performance. For downstream tasks, shown in Figure 3(c) we observe that all methods achieve the better performance already at $t = 1$, compared to the result of Class-IL. The overall results are upward-sloping, showing performance comparable to each of Joint. The experimental results of 6-Tasks and of downstream tasks for SupCon loss are proposed in S.M.

As a conclusion of this section, our findings in the supervised continual learning are summarized as follows:

- The problem of the biased prediction can be solved by simply re-training the output layer using the proposed GDumb-based exemplar memory.
- In Class-IL (relatively low similarity between tasks), both using the exemplar memory or the traditional algorithm (LWF and MAS) make a clear difference in learned representations. However, most of the existing CL algorithms including the current state-of-the-art method fail to accumulate more representations when using the exemplar memory.
• In Data-IL (relatively high similarity between tasks), existing applicable algorithms well accumulate representations than the scenario of Class-IL, resulting in the competitive performance in both the CL scenario and downstream tasks. However, the number of applicable algorithms is limited.

4.3 Experimental Analysis of Unsupervised Continual Learning in ImageNet-100

Experimental settings. Following the experiment on the supervised CL, we experiment with unsupervised (self-supervised) CL. Same as the setting of the supervised CL, we set three scenarios using the ImageNet-100 dataset. However, an input image is only used for training a model. We select Moco v2 \[12\] as an unsupervised (self-supervised) method for training an encoder of the model. The detailed settings and hyperparameters are proposed in S.M. Note that, we re-train the output layer using the GDumb-based exemplar memory $M_o$ after training the encoder with Moco v2, for evaluation in Class-IL and Data-IL.

![Figure 4: Experimental results of Class-IL in the unsupervised continual learning using ImageNet-100. CE and SupCon denote the used loss function, respectively. GD denotes the result after re-training the output layer with the GDumb-based exemplar memory $M_o$.](image)

Class-IL. Figure 4(a) shows experimental results of the unsupervised Class-IL. For the direct comparison with the result of the supervised CL, we plotted the representative results of it. From the figure, we observe two findings: First, the performance of Joint(Moco) surpasses any supervised class-IL methods including the state-of-the-art method (e.g., SS-IL) by a large margin. Second, when we use the size of over \(30k\) memory, FT(Moco) also beats the result of them. Note that, even though FT(Moco) needs over \(15\times\) more size of the exemplar memory, it only requires less than 2% of the supervised dataset (for maintaining $M_o$), compared to the supervised CL method which always requires the supervised datasets for training. Third, we applied some CL algorithms to Moco, but most of them results in worse performance, and only using the more exemplar memory improves performance. Subsequently, we evaluated each encoder, trained until the last task, on the downstream tasks and the averaged result is shown in Figure 4(b). Despite being trained only by the unsupervised datasets, we confirm that both Joint(Moco) and FT(Moco) with the enough size of memory achieve competitive performance compared to the results of the supervised CL method.

Data-IL. We conducted experiments of the unsupervised Data-IL and results are shown in Figure 5(a). We also plotted the best result of the supervised Data-IL in the figure. Moco(FT) with the enough size of memory achieved the competitive performance to CE(MAS) but it is a little short of the performance of IRD(SupCon). Experiments for downstream tasks, proposed in 5(b) show that the performance gap between the supervised and unsupervised CL becomes bigger than Class-IL. However, considering the labeling cost, we think Moco(FT) using enough size of exemplar memory achieves more than expected results.

Based on the above results, we summarized our findings on the unsupervised CL as below:

• Existing CL methods are not very effective for the unsupervised continual representation learning and only using the enough size of exemplar memory helps to accumulate representations.
Figure 5: Experimental results of Data-IL in the unsupervised continual learning using ImageNet-100. CE and SupCon denote the used loss function, respectively. GD denotes the result after re-training the output layer with the GDumb-based exemplar memory $\mathcal{M}_e$.

- We confirm that the unsupervised CL with the enough size of the exemplar memory can surpass or achieve a competitive performance compared to the supervised CL method, not only in both Class-IL and Data-IL, but also downstream tasks.
- In the real world, the cost of the size of memory is significantly cheaper than the labeling cost. Therefore, research on the unsupervised CL is both more realistic and promising as the future research direction of CL.

4.4 Experimental Analysis of Supervised Continual Learning in ImageNet-1000

Experimental settings We applied our analysis to the larger-scale dataset, ImageNet-1000 [15]. In the most cases, we followed the experimental setting used in the experiments for ImageNet-100 and consider two CL sequences, {100 classes × 10 tasks} (10-Tasks) and {500 classes, 10 classes × 5 tasks} (6-Tasks). We sampled 20 pairs of an input image and its target label per each class, for maintaining the GDumb-based exemplar memory $\mathcal{M}_e$. All detailed experimental settings are proposed in S.M.

Figure 6: Experimental results of the supervised class-incremental learning using ImageNet-1000. For all experiments, we applied the exemplar memory $|\mathcal{M}_e| = 20k$ except for Joint. CE and SupCon denote the used loss function, respectively. GD denotes the result after re-training the output layer with the GDumb-based exemplar memory $\mathcal{M}_e$.

Class-IL We applied several supervised Class-IL algorithms and the results are shown in Figure 6(a). Note that the result of more baselines is proposed in the S.M. As already proposed in their paper, SS-IL (CE) achieves the state-of-the-art performance among baselines. FT (CE) and MAS (CE) suffer from serious performance degradation as CL progresses. However, after re-training the output layer with Gdumb-based exemplar memory, we observe that their performance significantly increases. As a result, the performance difference between all baselines becomes minimized after re-training the output layer. By comparing the result of SS-IL (CE) and SS-IL (CE, GD), we again check that SS-IL's core contribution that diminishes the biased prediction in training time. However, when we
compare the results after re-training the output layer, the learned representations are not significantly different from the result of FT and rather like the result of MAS. In the case of SupCon-based methods, we observe that there are improvements after applying IRD but the performance is still lower than MAS(CE, GD). We conducted more experiments for other baseline algorithms and results are proposed in S.M.

Downstream tasks Figure 6(b) shows the experimental results of two downstream classification tasks. The results again demonstrate that the performance difference is insignificant between six methods, but the performance gap with Joint(CE) increases than the experiment of ImageNet-100. Additionally, as we already introduced in Section 4.1, we evaluated each trained encoder with the segmentation task. As shown in Figure 6(c), we confirm that there is a negligible performance difference between the trained encoders. However, we observe MAS(CE) achieves a slightly better result than SS-IL. We also did experiments for the unsupervised CL using the ImageNet-1000 dataset and all results are reported in S.M.

From the experiments on ImageNet-1000, we could again conclude that many CL algorithms do not stack representations well during CL. Especially, we observe that the biased prediction occurs more severely in ImageNet-1000. As a result, correcting the bias prediction allows the CL algorithm to achieve high accuracy even without accumulating representations in the CL process, making it mistaken for successful CL.

5 Concluding Remarks

We evaluated the major CL algorithms using the new evaluation criteria, re-training the output layer using class-balanced exemplar memory and testing on downstream tasks. By using them, we conducted extensive experiments using the ImageNet-scale dataset considering three CL scenarios. Our core findings can be summarized below.

Most major CL algorithms do not accumulate representations well As a result of re-training the output layer for the encoder trained in CL, we confirmed that most of the latest algorithms did not accumulate representations in CL. This tendency can be also observed in the experiments for downstream tasks. Based on the above experimental findings, we would like to argue that: 1) the evaluation method of the CL method should be diversified, 2) because retraining the output layer using the class-balanced memory can solve it easily, we should no longer concentrate on solving the biased prediction problem, 3) It should be pursued to study CL algorithms that can accumulate representations successfully.

Developing more efficient unsupervised CL algorithm We experimentally confirmed that the unsupervised CL using enough memory can outperform or achieve competitive performance compared to the supervised CL algorithms, in both the CL scenarios and downstream tasks. In this regard, we would like to suggest rethinking the cost of both memory and labeling in the scenario of CL. The memory cost is much cheaper than the labeling cost in the real world. Therefore, the study for the unsupervised CL with a small number of supervised data (for re-training the output layer) is much more realistic than the pure supervised CL. However, we found that only using enough memory helps to increase the quality of representations after the unsupervised CL. Therefore, we believe that devising a memory-efficient algorithm for the unsupervised CL is the most promising research direction for CL (e.g., a new regularization(or distillation) method and exemplar sampling policy for the unsupervised dataset).

6 Limitations

We believe our study has several limitations. First, our paper does not consider dynamic architecture-based methods. The reason that we have focused on the regularization(or distillation)-based method using an exemplar memory is because those methods have reported the most superior results in CL. However, many of the dynamic architecture-based methods only consider a small scale dataset and require tuning a lot of hyperparameters, making it hard to be applied to a new dataset. Nevertheless, we believe that it is necessary to evaluate whether such kinds of algorithms are stacking representations well. Second, we only consider Moco v2 [12] with naively chosen hyperparameters as the method for the unsupervised (self-supervised) CL. As we already know, more diverse self-supervised learning methods are proposed, such as SimCLR [11] and BarlowTwins [46]. Therefore, more experimental
studies, not only for finding the best setting of Moco v2 (e.g., hyperparameters and ResNet-18 model) but also for considering another self-supervised learning method, should be conducted in the future.

References

[1] Hongjoon Ahn, Sungmin Cha, Donggyu Lee, and Taesup Moon. Uncertainty-based continual learning with adaptive regularization. In Advances in Neural Information Processing Systems (NeurIPS), pages 4394–4404, 2019.

[2] Hongjoon Ahn, Jihwan Kwak, Subin Lim, Hyeonsu Bang, Hyojun Kim, and Taesup Moon. Ss-il: Separated softmax for incremental learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 844–853, 2021.

[3] Dosovitskiy Alexey, Philipp Fischer, Jost Tobias, Martin Riedmiller Springenberg, and Thomas Brox. Discriminative, unsupervised feature learning with exemplar convolutional, neural networks. IEEE TPAMI, 38(9):1734–1747, 2016.

[4] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In Proceedings of the European Conference on Computer Vision (ECCV), pages 139–154, 2018.

[5] Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 583–592, 2019.

[6] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In Proceedings of the European conference on computer vision (ECCV), pages 233–248, 2018.

[7] Hyuntak Cha, JaeHo Lee, and Jinwoo Shin. Co2l: Contrastive continual learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 9516–9525, 2021.

[8] Sungmin Cha, Hsiang Hsu, Taebaek Hwang, Flavio Calmon, and Taesup Moon. {CPR}: Classifier-projection regularization for continual learning. In International Conference on Learning Representations, 2021.

[9] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In Proceedings of the European Conference on Computer Vision (ECCV), pages 532–547, 2018.

[10] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc’Aurelio Ranzato. On tiny episodic memories in continual learning. arXiv preprint arXiv:1902.10486, 2019.

[11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In International conference on machine learning, pages 1597–1607. PMLR, 2020.

[12] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297, 2020.

[13] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In Proceedings of the fourteenth international conference on artificial intelligence and statistics, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[14] Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021.

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.

[16] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In Proceedings of the IEEE international conference on computer vision, pages 1422–1430, 2015.

[17] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16, pages 86–102. Springer, 2020.
[18] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[19] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06), volume 2, pages 1735–1742. IEEE, 2006.

[20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 9729–9738, 2020.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 831–839, 2019.

[22] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 831–839, 2019.

[23] Dapeng Hu, Shipeng Yan, Qizhengqiu Lu, Lanqing HONG, Hailin Hu, Yifan Zhang, Zhenguo Li, Xinchao Wang, and Jiashi Feng. How well does self-supervised pre-training perform with streaming data? In International Conference on Learning Representations, 2022.

[24] Ching-Yi Hung, Cheng-Hao Tu, Cheng-En Wu, Chien-Hung Chen, Yi-Ming Chan, and Chu-Song Chen. Compacting, picking and growing for unforgetting continual learning. Advances in Neural Information Processing Systems, 32, 2019.

[25] Sangwon Jung, Hongjoon Ahn, Sungmin Cha, and Taesup Moon. Continual learning with node-importance based adaptive group sparse regularization. In Advances in Neural Information Processing Systems (NeurIPS), volume 33, pages 3647–3658. Curran Associates, Inc., 2020.

[26] Divyam Madaan, Jaehong Yoon, Yuanchun Li, Yunxin Liu, and Sung Ju Hwang. Representational continuity for unsupervised continual learning. In International Conference on Learning Representations, 2022.

[27] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[28] German I Parisi, Ronald Kemker, Jose Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. Neural Networks, 113:54–71, 2019.

[29] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In European Conference on Computer Vision, pages 524–540. Springer, 2020.

[30] Dushyant Rao, Francesco Visin, Andrei Rusu, Razvan Pascanu, Yee Whye Teh, and Raia Hadsell. Continual unsupervised representation learning. Advances in Neural Information Processing Systems, 32, 2019.
[37] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 2001–2010, 2017.

[38] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. arXiv preprint arXiv:1606.04671, 2016.

[39] Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In International Conference on Machine Learning (ICML), pages 4528–4537, 2018.

[40] Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. arXiv preprint arXiv:1904.07734, 2019.

[41] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of machine learning research, 11(12), 2010.

[42] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[43] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In International Conference on Machine Learning, pages 9929–9939. PMLR, 2020.

[44] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 374–382, 2019.

[45] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In International Conference on Learning Representations (ICLR), 2018.

[46] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In International Conference on Machine Learning, pages 12310–12320. PMLR, 2021.

[47] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In European conference on computer vision, pages 649–666. Springer, 2016.

[48] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2881–2890, 2017.