IS CLASS INCREMENTAL LEARNING TRULY LEARNING REPRESENTATIONS CONTINUALLY?

Anonymous authors
Paper under double-blind review

ABSTRACT

Class incremental learning (CIL) aims to continually learn a classifier for new object classes from incrementally arriving data while not forgetting the past learned classes. The average test accuracy across all classes learned so far has been a widely used metric to evaluate the CIL algorithms, but we argue that a simple horse race toward maximizing the accuracy may not necessarily lead to developing effective CIL algorithms. Namely, since a classification model is often used as a backbone model that transfers the learned representations to other downstream tasks, we believe it is also important to ask whether the CIL algorithms are indeed *learning representations continually*. To that end, we borrow several typical evaluation protocols of representation learning to solely evaluate the quality of encoders learned by the CIL algorithms: 1) fix the encoder and re-train the final linear layer or run the k-nearest neighbor (NN) classifier using the entire training set obtained for all classes so far and check the test accuracy, and 2) perform transfer learning with the incrementally learned encoder to several downstream tasks and report the test accuracy on those tasks. Our comprehensive experimental results disclose the limitation of conventional accuracy-based CIL evaluation protocol as follows. First, the state-of-the-art CIL algorithms with high test accuracy do not necessarily perform equally well with respect to our representation-level evaluation, in fact, sometimes may perform even worse than naive baselines. Second, it turns out the high test accuracy of the state-of-the-art CIL algorithms may be largely due to the good quality of the representations learned from the *first* task, which means those algorithms mainly focus on stability (not forgetting the first task model's capability), but not really on continually learning new tasks, *i.e.,* plasticity, to attain high overall average accuracy. Based on these results, we claim that our representation-level evaluation should be an essential recipe for more objectively evaluating and effectively developing the CIL algorithms.

1 INTRODUCTION

In recent years, neural network has achieved great progress in various domains. However, such neural networks exhibit a large gap with humans in terms of their ability to continually learn from a series of tasks (Parisi et al., 2019; Delange et al., 2021; Masana et al., 2020). To narrow this gap in ability, the research field of continual learning (CL) has been considered a holy grail in neural network research. The ideal goal of CL is to successfully integrate knowledge gained from the new task (plasticity) while not forgetting knowledge of the previous tasks (stability) in the process of learning sequential tasks (Kirkpatrick et al., 2017; Ahn et al., 2019; Cha et al., 2021b). However, achieving this goal is difficult because neural networks suffer from a serious dilemma between stability and plasticity (Mermillod et al., 2013). To alleviate this problem, various types of CL algorithms have been proposed, categorized as follows: exemplar memory- (Chaudhry et al., 2019; Rebuffi et al., 2017), regularization- (Kirkpatrick et al., 2017; Aljundi et al., 2018; Chaudhry et al., 2018), and dynamic architecture-based (Schwarz et al., 2018; Rusu et al., 2016) algorithms.

Class incremental learning (CIL) (Masana et al., 2020) is an important sub-category of CL that has drawn a lot of attention recently. In CIL, a learning agent aims to continually learn a classifier for new object classes from incrementally arriving data while not forgetting the past learned classes. Such setting models the practical scenario that can be encountered in many real-world applications and is the hardest among different scenarios in CL (Van de Ven & Tolias, 2019), since the task-id

is not available at inference time. The effectiveness of the CIL algorithms are typically evaluated based on the average test 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 stability (for not forgetting past classes). Most of the recent CIL algorithms hence have been competitively proposed by aiming to increase the average test accuracy after learning the final task. For example, the regularization-based methods using the exemplar memory achieved the greatest progress in terms of test accuracy, even getting close to the performance of a model jointly trained with the entire training dataset (Ahn et al., 2021; Douillard et al., 2020; Kang et al., 2022).

We argue, however, that such a horse race toward maximizing the average test accuracy has limitations and may not necessarily lead to developing effective CIL algorithms. Our motivation comes from the fact that the classification models, when trained on a large-scale dataset (*e.g.*, ImageNet-1k), often serve as backbone models that transfer the learned representations (Kornblith et al., 2019) to diverse downstream tasks, *e.g.*, object detection or semantic segmentation. To that regard, we ask whether the CIL algorithms are indeed *learning representations continually* such that the incrementally learned classifiers may also serve as backbone models for transfer learning. With above reasoning, we borrow several typical evaluation protocols of representation learning to solely evaluate the quality of encoders learned by the CIL algorithms: 1) fix the encoder and re-train the final linear layer or run the k-nearest neighbor (NN) classifier using the entire training set obtained for all classes so far and check the test accuracy (to measure the representation quality for the learned tasks), and 2) perform transfer learning with the incrementally learned encoder to several downstream tasks and report the test accuracy on those tasks (to measure the generalizability of the learned representations). By testing with above representation-level evaluation protocol on class incrementally learning ImageNet-100 and ImageNet-1k, we obtained the following findings:

- To the best of our knowledge, this is the first comprehensive experimental analyses on the learned representations for existing state-of-the-art supervised CIL algorithms.
- The state-of-the-art CIL algorithms with high test accuracy do not necessarily perform well with respect to our representation-level evaluation, in fact, sometimes may perform even worse than naive baselines.
- It turns out the high test accuracy of the state-of-the-art CIL algorithms may be largely due to the good quality of the representations learned from the *first* task. Such factors affecting the final average accuracy of CIL were not properly considered in the conventional evaluation.

Based on our findings, we claim that our representation-level evaluation should be an essential recipe for more objectively evaluating and effectively developing the CIL algorithms.

2 Related Work

The type of CL methods is categorized into three different types (Delange et al., 2021). 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 (CIL) (Kirkpatrick et al., 2017; Aljundi et al., 2018; Chaudhry et al., 2018; Ahn et al., 2019; Jung et al., 2020; Mirzadeh et al., 2020; Cha et al., 2021b). Moreover, distillation-based methods can be considered as one subpart of it and focus on devising a distillation method that overcomes the catastrophic forgetting problem (Li & Hoiem, 2017; Douillard et al., 2020; Cha et al., 2021a).

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 (Rusu et al., 2016; Yoon et al., 2018; Mallya & Lazebnik, 2018; Schwarz et al., 2018; Hung et al., 2019; Lee et al., 2020). However, the weakness of those methods is known to be the complexity of the method, applicability to a large-scale dataset, and requirement of somewhat more hyperparameters.

Finally, exemplar-based methods were considered the most promising approach, demonstrating superior performance in most CIL scenarios Rebuffi et al. (2017); Castro et al. (2018); Wu et al. (2019); Hou et al. (2019); Prabhu et al. (2020); Ahn et al. (2021). To overcome the catastrophic forgetting,

it maintains a tiny exemplar memory saving a subset of the previous task's dataset and using them when the model is trained for a new task Chaudhry et al. (2019). However, because of the imbalance between the current and exemplar data in a mini-batch, the model encounters the *biased prediction* problem, in which the predictions are heavily biased toward the current task classes due to the severe data imbalance encountered in training time, and many methods are devised to alleviate it in CIL Wu et al. (2019); Hou et al. (2019); Ahn et al. (2021); Belouadah & Popescu (2019). As a result, they achieve the highest average accuracy on various datasets including the large-scale dataset (*e.g.* ImageNet) Ahn et al. (2021); Kang et al. (2022); Douillard et al. (2020).

3 Problem Formulation and Preliminaries

3.1 Problem formulation and scenarios of continual learning

Given a sequence of tasks, let $t \in \{1, \ldots, T\}$ represent the t^{th} task. Each task-specific dataset $D_t = \{(\boldsymbol{x}_t^{(i)}, y_t^{(i)})\}_{i=1}^{N}$ consists of N pairs of an input image and its target label. We assume that y_t is sampled from a task-specific class set $\mathcal{C}_t$ such that $y_t \in \mathcal{C}_t$. We train a classification model $f_{\boldsymbol{\theta}} = (h_{\boldsymbol{\psi}} \circ g_{\boldsymbol{\phi}})$, where $h_{\boldsymbol{\psi}}$ and $g_{\boldsymbol{\phi}}$ denotes an encoder and output layer of the model, respectively. More specifically, at task t, $f_{\boldsymbol{\theta}_t}$ is trained on D_t for multiple epochs (offline training) without having access to any other datasets, after which then t increments by one. An exemplar memory $\mathcal{M}_e$ is often employed to store and replay a small number of data instances from previously seen tasks. By leveraging the target labels, we can balance the class distribution of the samples stored in $\mathcal{M}_e$, and these class-balanced samples can be interleaved with D_t to train the model $f_{\boldsymbol{\theta}}$. For the cross-entropy (CE) objective function, both the encoder and output layer are trained jointly. Note that we denote a model trained with the entire training datasets until task t as the joint model.

Class- and task-incremental learning In these scenarios, each class set $\mathcal{C}_t$ has disjoint class labels: $\mathcal{C}_j \cap \mathcal{C}_k = \emptyset$, $\forall j, k \in \{1, \ldots, T\}$ and $j \neq k$. At inference time, task-incremental learning (TIL) provides an additional supervisory signal that indicates the task-id of an input image. The TIL makes it straightforward to select a dedicated output layer for each task learned during training. The resulting multi-head configuration exhibits less interference between different tasks. Meanwhile, class-incremental learning (CIL) requires no such additional supervisory signals during inference, as it adopts a shared output layer. Its single-head configuration, however, causes CIL to be more prone to catastrophic forgetting than TIL. The regularization-(or distillation-)based algorithms using the exemplar memory have achieved a highly successful result in the large-scale dataset (e.g., ImageNet dataset), even significantly close to the performance of the joint model (Rebuffi et al., 2017; Cermelli et al., 2020; Hou et al., 2019; Ahn et al., 2021; Kang et al., 2022; Douillard et al., 2020; Masana et al., 2020).

4 Representation-level Evaluation for CIL

Throughout the achievement of CIL research, average test accuracy-based evaluation metrics, including the final test accuracy, average test accuracy, forgetting, and intransigence measures (Chaudhry et al., 2018; Cha et al., 2021b), have been considered as standard metrics. However, as mentioned in the Introduction, we raise the following question: *Is achieving high test accuracy sufficient for developing good CIL algorithms?* The ultimate goal of CIL can be regarded as attaining a model that works similarly as the joint model that is trained with full training data observed so far. However, since the evaluation metric for '*similar*' is ambiguous, most studies aim to follow the classification performance of the joint model as closely as possible (Delange et al., 2021; Masana et al., 2020), especially in class-IL. On the other hand, the joint model (*e.g.*, ImageNet pre-trained model) is not only used for classification, but is also actively applied to various downstream tasks for transfer learning. This implies that the value of the joint model is not only in the high average classification accuracy but also in learning general representations that can be transferred to downstream tasks (Kornblith et al., 2019). Similarly, the value of a CIL algorithm can be also evaluated by the quality of the representations it learned during the CIL process, but this angle has been largely neglected in evaluating the CIL algorithms. In this regard, we carry out comprehensive evaluations on the quality of the representations learned by the state-of-the-art CIL algorithms.

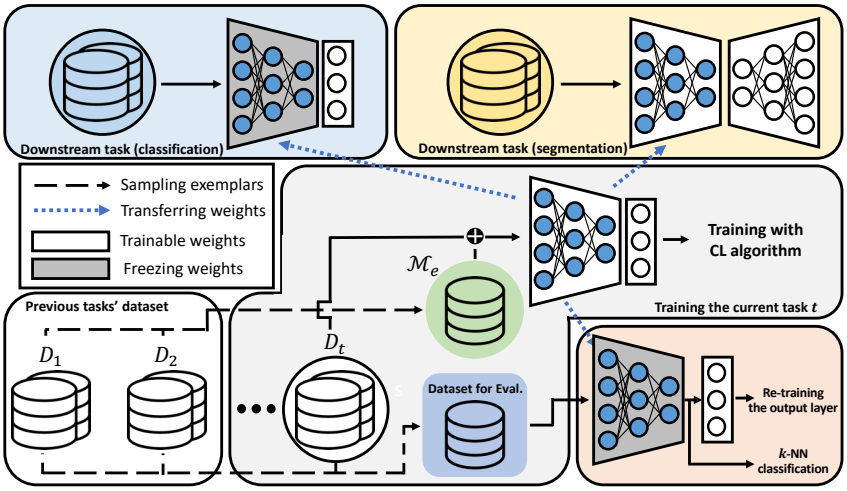


Figure 1: Illustration of the our proposed evaluation protocols.

More concretely, we first report the ordinary average accuracy of the classifier $f_{\boldsymbol{\theta}_t}$ on the test sets for the classes from task 1 to t after learning each task t, denoted as [**Acc**$_{(t)}$]. Then, we borrow several typical evaluations protocols of representation learning (Zbontar et al., 2021; He et al., 2020) and evaluate the encoders, $h_{\boldsymbol{\psi}_t}$, of each CIL method as depicted in Figure 1. We carry out two types of the representation-level evaluations as described below.

(a) **Evaluating with the in-domain dataset** We first evaluate the learned representations of each encoder $h_{\boldsymbol{\psi}_t}$ with the test dataset for CIL tasks itself (*e.g.*, ImageNet-100 or ImageNet-1000). That is, at the end of each training task t, we freeze the encoder $h_{\boldsymbol{\psi}_t}$ and re-train the final linear layer or run the k-NN classifier ($k = 20$) using the *entire* training set for task 1 to t and report the accuracy on the test sets also from task 1 to t. We denote those accuracy as [**Linear**$_{(t)}$] and [k-**NN**$_{(t)}$], respectively. Moreover, we also report [**Linear**$_{(t,T)}$] and [k-**NN**$_{(t,T)}$] for $t \neq T$, which stand for the accuracy results of the t-th encoder for the entire test datasets (from task 1 to T) by re-training the output layer or by running the k-NN classifier with the entire training dataset (from task 1 to T).

(b) **Evaluating with the out-domain datasets** To evaluate the generalizability of the learned representations of the encoder, we conduct experiments of transfer learning with out-domain datasets as well. In the case of CIL with ImageNet-100, we consider three downstream tasks of classification, namely STL-10 (Coates et al., 2011), CUB200 (Wah et al., 2011), and resized CIFAR-10 (Krizhevsky et al., 2009). For each encoder $h_{\boldsymbol{\psi}_t}$, we perform linear evaluation using each dataset and report their average classification accuracy of them, denoted as [**CLS**$_{(t)}$]. For CIL with the ImageNet-1k dataset, we conduct more diverse experiments including the above experiments. That is, we perform linear evaluation of each encoder $h_{\boldsymbol{\psi}_t}$ for both the multi-label classification task using VOC 2012 dataset (Everingham et al.) (denoted as [**MLC**$_{(t)}$]) and the classification task using the iNaturalist dataset (Van Horn et al., 2018) (denoted as [**iNat**$_{(t)}$]). Furthermore, we use each encoder $h_{\boldsymbol{\psi}_t}$ as the initialization for an encoder of PSPNet (Zhao et al., 2017) and train the entire model of PSPNet with the VOC 2012 semantic segmentation dataset (Everingham et al.), without freezing the encoder. The resulting semantic segmentation performance in terms of mean Intersection-over-Union (IoU) is denoted as [**Seg**$_{(t)}$].

5 Experimental Results

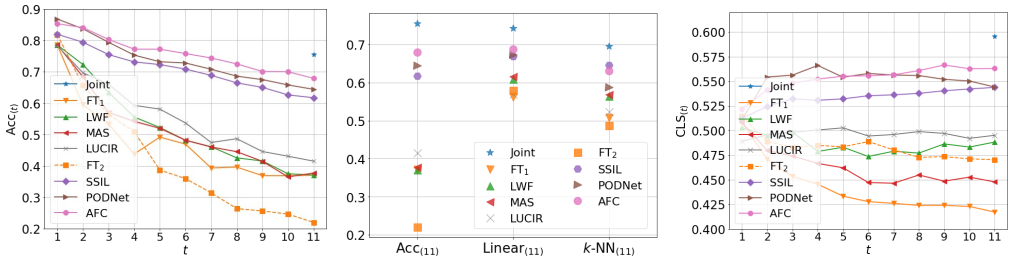
5.1 Experiments with ImageNet-100

The ImageNet-100 dataset has been used as one of the popular datasets to evaluate CIL algorithms Douillard et al. (2020); Hou et al. (2019); Kang et al. (2022); Wu et al. (2019). We accordingly conducted CIL experiments with the ImageNet-100 dataset and analyze the experimental results with the proposed evaluations. We consider three different CIL scenarios for the experiments. First, a scenario starting from learning a base task (consisting of 50 classes) and evenly learning remaining classes for 10 tasks (denoted as 11-tasks) with the exemplar memory. Second, a scenario

sequentially learning each of 10 tasks consisting of 10 classes (denoted as 10-tasks) with the exemplar memory. The third setting is identical with the 10-tasks scenario but further examine with the CIL methods *without* the exemplar memory (proposed in the S.M.).

Baselines We consider several baselines including representative regularization-(or distillation-) based algorithms, such as MAS (Aljundi et al., 2018) and LWF (Li & Hoiem, 2017), and state-of-the-art (SOTA) CIL algorithms, such as LUCIR (Hou et al., 2019), SSIL (Ahn et al., 2021), PODNet (Douillard et al., 2020), and AFC (Kang et al., 2022). We denote the results of fine-tuning and the joint model as FT_1 and Joint, respectively. Note that we obtain the results of Joint, FT_1, MAS, LUCIR and LWF, by implementing the CIL framework proposed by (Masana et al., 2020), while the results of SSIL, PODNet, and AFC are reproduced by running their official code, without any modification of the hyperparameters and configurations. We also implement another version of fine-tuning, denoted as FT_2, which used the code of Ahn et al. (2021). Here, the difference with FT_1 is to train 1/4 of the total epochs, save for the first task, to prevent over-fitting to the new task. We used ResNet-18 (He et al., 2016) as a classifier model and reported Top-1 accuracy for all experiments. As a default setting, we applied the exemplar memory $|\mathcal{M}_e| = 2000$ for all baselines. All proposed evaluations are conducted using the modified CIL framework code. Detailed settings and hyperparameters are introduced in the S.M.

Class-incremental learning with exemplars (11-tasks with base task) Recently, the CIL scenario beginning from a base task, which learns a half of the entire classes as the first task, has become a popular setting for CIL research (Douillard et al., 2020; Hou et al., 2019; Kang et al., 2022). We first conduct the experimental analyses in this scenario, in which 50 classes are learned as the first task and then 10 tasks (5 classes for each task) are incrementally learned.



(a) In-domain eval. ($\text{Acc}_{(t)}$). (b) In-domain evaluations. (c) Out-domain eval. ($\text{CLS}_{(t)}$).

Figure 2: Experimental results of the state-of-the-arts CIL algorithms using the exemplar memory $|\mathcal{M}| = 2000$ for the scenario of 11-tasks with the base task.

Figure 2(a) shows $\textbf{Acc}_{(t)}$ results, and we observe that FT_1/FT_2, LWF, and MAS suffer from worse performance, primarily due to the biased prediction problem as reported in (Belouadah & Popescu, 2019; Wu et al., 2019). On the other hand, the final task's performance of three SOTA algorithms (SSIL, PODNet, and AFC) are getting closer to Joint than other baselines, demonstrating their superior performance in terms of $\textbf{Acc}_{(t)}$. The accuracy gap between those three and other baselines is significant, roughly up to 48%. However, the representation-level evaluation results of $\textbf{Linear}_{(11)}$ and k-$\textbf{NN}_{(11)}$ in Figure 2(b) present that the performance gap among all baselines significantly closes unlike in $\textbf{Acc}_{(t)}$ results. Particularly for $\textbf{Linear}_{(11)}$, we observe the top 3 SOTA baselines only obtain about $+10\%$ better accuracy than the naive FT baselines. We believe these results show that the representation quality of the SOTA CIL methods and naive baselines are in fact *not* drastically different, and the large gap in $\textbf{Acc}_{(t)}$ may be primarily due to the advanced mechanisms for addressing the biased prediction issue. Furthermore, Figure 2(c) depicts the transfer learning experimental results for the out-domain datasets. In the process of learning tasks continuously, we confirm that only SOTA baselines show the right-upward performance but others are not, appearing to be learning the representation continuously well.

From the experiments in Figure 2(c), the SOTA baselines seem to continually learn representations well in this scenario. Here, the remaining question is: How much do the SOTA algorithms learn better representations? To evaluate these algorithm more exactly, we conduct the evaluation with both $\textbf{Linear}_{(1,11)}$ and $\textbf{Linear}_{(11,11)}$, shown in Figure 3. Note that the first task's model of each method is only trained with the cross-entropy loss function. The figure presents that, first, because of the difference in the hyperparameters and configuration of each algorithm (*e.g.*, learning rate and types

of classifier), the representation quality learned at the base task is different from each other (ranging from about 60% to 64%). For a fair comparison, we perform the CIL experiment by unifying the base model of SSIL with the first task model of LWF, and the result is denoted as SSIL* in the figure. As a result of starting CIL from the base model that achieves 2% lower performance (the LWF's base model), we obtain a final performance that is 1% lower performance than before. This experimental result demonstrates that the performance of the base model affects the final performance. Note that we could not conduct the same experiment for both PODNet and AFC since these methods use a cosine classifier but LWF and SSIL are not. Second, from the performance gap between $\textbf{Linear}_{(1,11)}$ of each SOTA algorithm and of Joint (about $10-12\%$), we confirm that this CIL scenario already start from superior representations at the base task. This implies that algorithms that take more stability into account may be advantageous in this scenario. Third, the performance difference between $\textbf{Linear}_{(11,11)}$ and $\textbf{Linear}_{(1,11)}$ can be considered as a true gain of each algorithm in representations. Among three SOTA baselines, SSIL rather achieves better improvement (about 5% of accuracy) than others but the performance gain of the SOTA baselines (difference between $\textbf{Linear}_{(11,11)}$ and $\textbf{Linear}_{(1,11)}$) is not significantly different (within $4-5\%$).

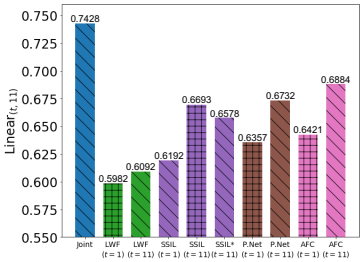
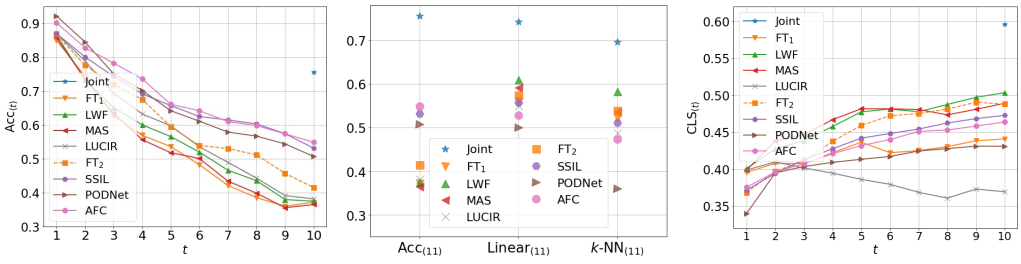


Figure 3: $\text{Linear}_{(t,11)}$

From the experimental results of 11-tasks, we conclude that:

- First, the actual difference in the representation quality is not significant than the reported CIL accuracy.
- Second, the quality of learned representations at the base task is already superior and diverse from each other. Therefore, considering the performance difference between the base and final model is more suitable to evaluate each algorithm appropriately.
- Finally, the CIL scenario starting from learning the base task is more favor with an algorithm considering stability more. Namely, this scenario is not ideal from the perspective of evaluating stability and plasticity of CL algorithm at the same time.

Class-incremental learning with exemplars (10-tasks) As an another representative scenario of CIL, we conduct experiments in the scenario that learns 10 classes for 10 tasks continually.



(a) In-domain eval. ($\text{Acc}_{(t)}$). (b) In-domain evaluations. (c) Out-domain eval. ($\text{CLS}_{(t)}$).

Figure 4: Experimental results of the state-of-the-arts CIL algorithms using the exemplar memory $|\mathcal{M}_e| = 2000$ in the 10-tasks scenario.

Figure 4 depicts the experimental results in this scenario and Figure 4(a) represents the result of $\text{Acc}_{(t)}$ of each baseline. FT_2 yields relatively better performance than FT_1 because if suffers less from biased prediction by reducing the training epochs. The current SOTA algorithms, including SSIL, PODNet, and AFC, exhibit a large performance gap with respect to other baselines in this scenario again, demonstrating an accuracy difference of up to 26%. Next, we performed the proposed evaluations using the encoders trained by each algorithm, shown in Figure 4(b). From both evaluations, $\textbf{Linear}_{(10)}$ and $\textbf{KNN}_{(10)}$, we observe that the order of performance is totally reversed compared with $\textbf{Acc}_{(10)}$. In particular, we confirm that not only LWF surpasses the SOTA algorithms in both evaluations, but also these SOTA algorithms even

achieve worse performance than both FT_1 and FT_2 as well. Figure 4(c) depicts the experimental results of the linear evaluation for three downstream tasks (out-domain datasets). Based on the upward trend of all the experimental results except for LUCIR, we confirm that most algorithms learn a gradually improved representations throughout the CL process, even without using the CIL algorithm (FT). However, as we already observed in previous experiments, the SOTA algorithms exhibit not only inferior final performance than FT_2, LWF and MAS, but also achieve less performance improvement when considering the difference between $\textbf{CLS}_{(10)}$ and $\textbf{CLS}_{(1)}$ (*e.g.*, about 10% improvement of SSIL, but 12% improvement of FT_2 and 20% improvement of LWF). Additionally, in the scenario of 10-tasks, we observe that LUCIR constantly shows worse results in all evaluations.

Finally, Figure 5 presents a bar graph plotting $\textbf{Linear}_{(1,10)}$ and $\textbf{Linear}_{(10,10)}$ of four representative algorithms. Different from the result of the 11-tasks scenario, we observe that LWF learns slightly better representations at the first task than other baselines. Nevertheless, when considering the performance difference between $\textbf{Linear}_{(10,10)}$ and $\textbf{Linear}_{(1,10)}$, we again confirm that LWF yields the greatest gain (over $+30\%$ of accuracy) compared with other SOTA baselines (less than $+25\%$ of accuracy).

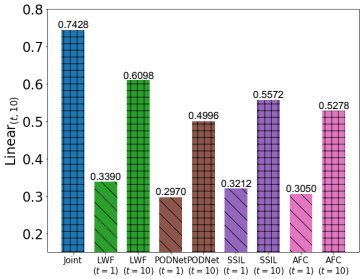


Figure 5: $\text{Linear}_{(t)}$.

Based on the experimental results in the CIL scenario of 10-tasks, we can say that:

- First, while the SOTA algorithms appear to achieve excellent performance in the general evaluation metric ($\text{Acc}_{(t)}$), but they may have learned worse representations than other baselines (*e.g.*, FT and LWF) in the 10-tasks scenario.
- Second, LWF learns the most superior representation during the 10-tasks scenario, but appears to achieve poor results in the general metric ($\text{Acc}_{(t)}$) due to severe biased prediction seriously.
- Third, the representation quality of the first task's model can be slightly different in this scenario also.

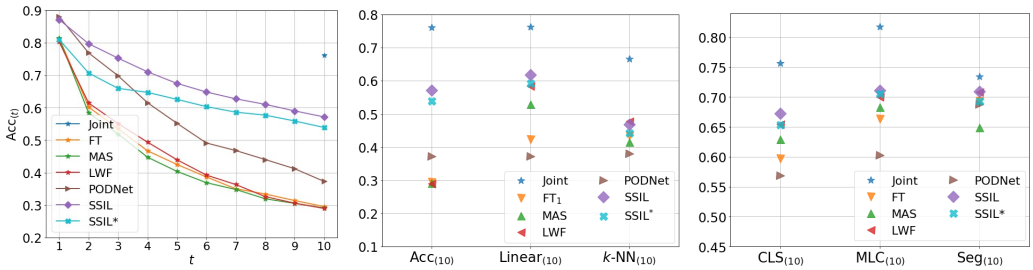
In the S.M, we conduct additional experiments of CIL without using the exemplars. Here, we confirm that the traditional regularization-based methods (*e.g.*, LWF and MAS) make a great improvement in representations. However, most SOTA CIL algorithms do not effectively use the exemplars in terms of learning better representations.

5.2 Experiments with ImageNet-1k

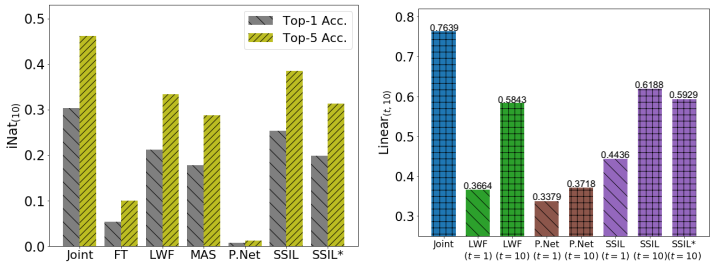
The experimental result for a large-scale dataset have been considered as the most important factor. Many studies verified the superiority of their algorithm through experiments on the ImageNet-k dataset (Wu et al., 2019; Ahn et al., 2021; Kang et al., 2022; Hou et al., 2019; Douillard et al., 2020). In this section, we conduct the same experimental analyses in order to verify whether the reported performance of each algorithm on the ImageNet-1k dataset truly represents learned representations by them. We focus on the 10-tasks scenario of CIL (sequentially learning 10 tasks consisting of 100 classes, respectively), which turned out to be the more suitable scenario for the sake of comparing both plasticity and stability of CIL algorithm in terms of the representation quality.

Experimental settings We selected some representative algorithms, such as FT, MAS, LWF, PODNet, and SSIL. We maintain most settings, used in the experiments for ImageNet-100, but changed the followings: First, we used ResNet-50 (He et al., 2016) as the classifier model. Second, we obtain the experimental result of PODNet by means of implementing the reproduce in the official code of SSIL. Note that we are. unable to run the official code of both AFC and PODNet given that these return errors, during the experiments for the scenario of 10-tasks with the ImageNet-1k dataset. We apply the exemplar memory $|\mathcal{M}_e| = 20000$ for all experiments. More details covering the hyperparameters used in the experiments of downstream tasks are proposed in the S.M.

Class-incremental learning with exemplars (10-tasks) As we already presented in Section 4, we conducted more diverse analyses for the experiment with the ImageNet-1k, as can be seen in Figure



(a) In-domain eval. ($\text{Acc}_{(t)}$). (b) In-domain evaluations. (c) Out-domain evaluations.



(d) Out-domain eval. ($\text{iNat}_{(10)}$) (e) In-domain eval. ($\text{Linear}_{(t,10)}$)

Figure 6: Experimental results of the state-of-the-arts CIL algorithms using the exemplar memory $|\mathcal{M}_e| = 20000$ in the scenario of 10-tasks with base task.

6. Figure 6(a) presents the $\textbf{Acc}_{(t)}$ of each algorithm. As already proposed in the SSIL paper (Ahn et al., 2021), we confirm that SSIL achieves the SOTA performance in this scenario and PODNet reveals its weakness in the case where the base task is not considered in the CIL scenario. Additionally, other baselines exhibit worse performance due to serious biased predictions, as reported in several papers (Wu et al., 2019; Belouadah & Popescu, 2019; Ahn et al., 2021). Next, we conducted the experiments with $\text{Linear}_{(10)}$ and $k\textbf{-NN}_{(10)}$ and Figure 6(b) depicts the results. Compared with the results depicted in Figure 4(b), we confirm similar but slightly varying tendencies. First, except for PODNet, all baselines achieve increased accuracy in both evaluations as compared to $\textbf{Acc}_{(10)}$. Second, the performance difference between LWF and SSIL decreases in $\textbf{Linear}_{(10)}$ but SSIL surpasses LWF, by a small margin.

Figure 6(c) and 6(d) present the experimental results for out-domain datasets. For all experiments, we again observe a similar tendency: First, SSIL and LWF produce a similar performance in most cases, but SSIL is more superior in $\textbf{iNat}_{(10)}$ to a greater degree, as depicted in Figure 6(d). Second, PODNet consistently achieves inferior performance in most downstream tasks. Third, the performance difference in $\textbf{Seg}_{(10)}$ between baselines is almost negligible but MAS shows a noticeably worse result. Based on the above results, we once confirm that an order of the representation quality learned by each CIL algorithm can be different from the performance order of $\textbf{Acc}_{(10)}$. Also, three additional analyses, which are suitable for a model trained by the ImageNet-1k dataset, more dramatically demonstrate the representation quality learned by each algorithm, and their tendency of them is similar to the findings in the experiments using the ImageNet-100 dataset.

However, one remaining question is that, different from the experiment using the ImageNet-100 dataset, SSIL constantly achieves better performance in most evaluations. To compare it more exactly, we conducted experiments of $\textbf{Linear}_{(t,10)}$ for three baselines and results are presented in Figure 6(e). When comparing $\textbf{Linear}_{(10,10)}$ with $\textbf{Linear}_{(1,10)}$, LWF shows the best improvement with about $+22\%$ improvement of accuracy, larger than SSIL with about $+17\%$ improvement. Also, we confirm that the first task's representation quality can be different from each other in the 10-tasks scenario with the ImageNet-1k dataset. Notably, the difference of $\textbf{Linear}_{(t,10)}$ between SSIL and LWF is about 9%, and we believe that it might significantly affect not only the CIL accuracy ($\textbf{Acc}_{(t)}$) but also the representation quality learned by CIL algorithm. For a fair comparison, we additionally implemented SSIL starting from the LWF's first task model and conducted the same experimental analyses, and we denote the results as SSIL* in Figure 6. The experimental results again verify that: First, the representation quality of the first task's model significantly affects the result of the entire

evaluation. After unifying the first task's model, the performance of both methods become almost similar in most evaluation except for $\mathbf{Acc}_{(t)}$. Second, the SOTA algorithm (SSIL) does not learn better representations than LWF, also in the CIL with the ImageNet-1k dataset.

In conclusion, we can summarize additional findings from the experiments with the ImageNet-1000 dataset as follows:

- First, the SOTA CIL method with high accuracy dose not learn better representations in the CIL with the large-scale dataset.
- Second, the additional evaluation for the CIL scenario with the ImageNet-1000 dataset more dramatically demonstrates the representation quality learned by each CIL algorithm.
- Third, the representation quality of the first task's model can affect the evaluation for CIL algorithm, even in the 10-tasks scenario.

6 Concluding Remarks

We propose to rethink the current tendency of class-incremental learning (CIL) research toward maximizing the average test accuracy. Based on the fact that using a pre-trained classification model in diverse downstream tasks, we argue that the goal to be pursued in CIL research is not simply to achieve a high average test accuracy, but to learn superior representations continually. In this regard, to the best of our knowledge, we propose comprehensive experimental analyses of the representation quality learned by each state-of-the-art CIL algorithm. From the extensive experiments using both the ImageNet-100 and ImageNet-1k datasets, we confirm that: First, the average test accuracy of CIL does not represent the representation quality learned by CIL algorithm. Second, existing state-of-the-art CIL algorithms might not be able to learn better representations than a naive approach. Third, the representation quality of the first task's model, which is trained by cross-entropy loss only, depends on the hyperparameter settings of CL algorithm, and the test accuracy of the state-of-the-art CIL algorithms tends to be highly influenced by it.

Based on the above findings, we would like to make the following suggestions for objective evaluation of CIL algorithm and future CL research:

- First, the evaluation of CIL algorithm should be more diversified with appropriate evaluation. In our opinion, representation-level evaluation is more objective.
- Second, the representation quality of the first task's model can affect overall evaluation for CIL algorithm. It is more fairer to unify the first task's model or to consider the performance difference between $\text{Linear}_{(T,T)}$ and $\text{Linear}_{(1,T)}$.
- Third, the CIL scenario starting from learning the base task is more in favor of an algorithm considering stability. Therefore, the experiments should be conducted in various scenarios to evaluate both stability and plasticity of the CIL algorithm well.
- Finally, most SOTA CIL algorithms tend to focus on maintaining stability more than plasticity. In this regard, we believe that devising a CIL algorithm that can learn representation continuously is more promising.

REFERENCES

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

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*, pp. 844–853, 2021.

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)*, pp. 139–154, 2018.

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

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)*, pp. 233–248, 2018.

Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9233–9242, 2020.

Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2l: Contrastive continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9516–9525, 2021a.

Sungmin Cha, Hsiang Hsu, Taebaek Hwang, Flavio Calmon, and Taesup Moon. {CPR}: Classifier-projection regularization for continual learning. In *International Conference on Learning Representations*, 2021b. URL `https://openreview.net/forum?id=F2v4aqEL6ze`.

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)*, pp. 532–547, 2018.

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.

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*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

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.

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*, pp. 86–102. Springer, 2020.

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.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

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*, pp. 9729–9738, 2020.

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*, pp. 831–839, 2019.

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.

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, pp. 3647–3658. Curran Associates, Inc., 2020.

Minsoo Kang, Jaeyoo Park, and Bohyung Han. Class-incremental learning by knowledge distillation with adaptive feature consolidation. *arXiv preprint arXiv:2204.00895*, 2022.

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.

Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2661–2671, 2019.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim. A neural dirichlet process mixture model for task-free continual learning. *arXiv preprint arXiv:2001.00689*, 2020.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

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.

Marc Masana, Xialei Liu, Bartlomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost van de Weijer. Class-incremental learning: survey and performance evaluation on image classification. *arXiv preprint arXiv:2010.15277*, 2020.

Martial Mermillod, Aurélia Bugaiska, and Patrick Bonin. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects. *Frontiers in psychology*, 4:504, 2013.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh. Understanding the role of training regimes in continual learning. *Advances in Neural Information Processing Systems*, 33:7308–7320, 2020.

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

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

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*, pp. 524–540. Springer, 2020.

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*, pp. 2001–2010, 2017.

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.

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)*, pp. 4528–4537, 2018.

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

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8769–8778, 2018.

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

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*, pp. 374–382, 2019.

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

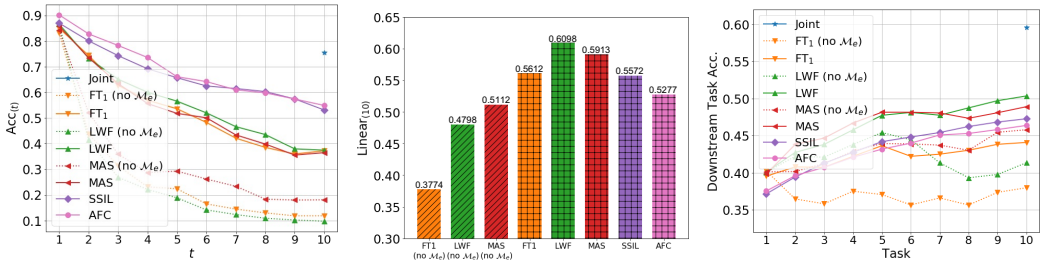
Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pp. 12310–12320. PMLR, 2021.

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*, pp. 2881–2890, 2017.

A APPENDIX

A.1 ADDITIONAL EXPERIMENTAL RESULTS WITH THE IMAGENET-100 DATASET

Class-incremental learning without the exemplars (10-tasks) Due of the difficulty of CIL compared with that of other scenarios, using the exemplar memory has become the basic setting for the state-of-the-art regularization-(or distillation-)based methods (Wu et al., 2019; Hou et al., 2019; Belouadah & Popescu, 2019; Douillard et al., 2020; Ahn et al., 2021; Rebuffi et al., 2017; Kang et al., 2022). In the majority of CIL algorithms, the exemplars are used as a direct exemplar of previous tasks in order to overcome catastrophic forgetting Douillard et al. (2020); Kang et al. (2022) or as a clue for alleviating biased predictions Wu et al. (2019); Belouadah & Popescu (2019); Ahn et al. (2021). However, there are unclear points: in terms of the quality of learned representations, the impact of utilizing the exemplar and whether the SOTA algorithms are using the exemplar efficiently. To address the above questions, we conducted additional experiments for the 10-tasks scenario without using the exemplar memory. Note that the SOTA baselines, SSIL and AFC, are designed based on leveraging the exemplar. Therefore, we only implement both LWF and MAS without using the exemplar, denoted as (no $\mathcal{M}_e$), and compare with other baselines.



(a) In-domain eval. ($\text{Acc}_{(t)}$) (b) In-domain eval. ($\text{Linear}_{(10)}$) (c) Out-domain eval. ($\text{CLS}_{(t)}$)

Figure 7: Experimental results of the state-of-the-arts CIL algorithms with and without the exemplar memory $|\mathcal{M}| = 2000$ in the 10-tasks scenario. Note that (no $\mathcal{M}_e$) denotes the case that the exemplar memory is not applied.

Figure 7 presents the experimental results on the cases of absent using the exemplar. Note that we attached the results of Joint, FT_1, LWF, MAS, SSIL, and MAS applying the exemplar for comparison. From Figure 7(a), we confirm that using the exemplar helps in the achievement of better performance in the $\text{Acc}_{(t)}$ metric (maximum about $+30\%$ improvement of accuracy). Figure 7(b) compares $\text{Linear}_{(10)}$ of each baseline as the comparison of the quality of learned representations. From this figure, we can derive several observations: First, when an algorithm is not applied (*e.g.*, FT), using the tiny exemplar significantly increases the representations quality ($+22\%$ improvement in accuracy). Second, both MAS and LWF allow a model to learn better representations efficiently (maximum $+15\%$ improvement of accuracy than FT_1), in the case of no use of the exemplar. Note that these results are not revealed in the $\text{Acc}_{(t)}$ metric. Third, in the cases of with and without the exemplar for both LWF and MAS, LWF more successfully utilizes the exemplars than does MAS. This is accomplished by achieving a 17% greater improvement in accuracy. Finally, the result of both SSIL and AFC presents that they have learned, not only slightly inferior representations than the fine-tuning, but also slightly better representations than MAS without the exemplar. Figure 7(c) depicts the experimental results on out-domain datasets and makes clear that both SOTA algorithms learn the similar quality of representations with MAS (no $\mathcal{M}_e$) and FT_1.

Namely, these experimental results demonstrates followings:

- First, both LWF and MAS clearly work as learning better representations in CIL when the exemplar is not applied. Also, in the case of using the exemplars, LWF most effectively utilizes them for learning the better quality of representations.
- Second, even though $\text{Acc}_{(t)}$ is higher than other baselines, both SOTA algorithms are not efficiently utilizing the exemplars in terms of learning better representations.

A.2 THE HYPERPARAMETERS AND EXPERIMENTAL SETTINGS

A.2.1 CLASS-INCREMENTAL LEARNING ALGORITHM

Experiments with ImageNet-100 For the cases of FT_1, MAS (Aljundi et al., 2018), LWF (Li & Hoiem, 2017) and LUCIR (Hou et al., 2019), we achieve the result by implementing the CIL framwork code proposed by (Masana et al., 2020), without any modification of default value of each algorithm's hyperparameter. We trained these algorithms for 100 epochs for each task and used SGD optimizer with 0.1 of the initial learning rate, 0.9 of momentum and 0.0001 of weight decay. We set the learning rate schedule which drop the learning rate by $\times 0.1$ at 40 and 80 epochs, respectively. Mini-batch size is set to 256 for all experiments. As the sampling algorithm for the exemplar memory, we used random sampling.

The SOTA CIL algorithms, such as PODNet Douillard et al. (2020), SSIL Ahn et al. (2021) and AFC Kang et al. (2022), are implemented by running their official code, also without any modification of not only defalut values of the hyperparameters but also other settings for training (*e.g.*, learning rate, epochs and mini-batch size).

Experiments with ImageNet-1k We experimented with FT_1, MAS and LWF by running the CIL framework code proposed by (Masana et al., 2020). Note that we trained the above baselines with the same settings used in the experiment of the ImageNet-100 dataset.

We tried to implement all SOTA algorithms in the 10-tasks scenario using the ImageNet-1k dataset but the official code of both PODNet and AFC returns error (they share the equal code base), so we could not achieve the results of them. Alternatively, we run the reproduced version of PODNet and SSIL, implemented in the official code of SSIL. Again, we maintained all hyperparameters and settings for training without any modification.

A.2.2 IN-DOMAIN EVALUATION

Linear evaluation We re-trained an output layer only while freezing an encoder. We trained the output layer for 100 epochs using 256 of mini-batch size, and used SGD optimizer with 0.1 of the initial learning rate, 0.9 of momentum and 0.0001 of weight decay. We set the learning rate schedule which drop the learning rate by $\times 0.1$ at 40 and 80 epochs, respectively.

k-NN evaluation We run k-NN implementation ($k = 20$) of Scikit-learn (Pedregosa et al., 2011) for all experiments. For classification, we first fit k-NN using the outputs of an encoder for given inputs. Then, we classify a given test data with the k-NN classifier.

A.2.3 OUT-DOMAIN EVALUATION

Three downstream tasks of classification We selected CIFAR-10 Krizhevsky et al. (2009), STL-10 Coates et al. (2011) and CUB-200 Wah et al. (2011) as downstream tasks for out-domain evaluation. For CIFAR-10, we randomly selected 5,000 training images from the entire training dataset and used a resized input image to 96×96. In the cases of STL-10 and CUB-200, we used the entire training dataset and maintain its original image size. Then, we only trained a newly added output layer while freezing an encoder. For CIFAR-10 and CUB-200, we trained the output layer for 100 epochs using 128 of mini-batch size, and used SGD optimizer with 0.1 of the initial learning rate, 0.9 of momentum and 0.0001 of weight decay. We set the learning rate schedule which drop the learning rate by $\times 0.1$ at 40 and 80 epochs, respectively. On the other hand, we changed the number of epochs and initial learning rate into 10 and 0.005, respectively.

Multi-label classification using VOC 2012 dataset For multi-label classification, we also trained an output layer only. Before training, we attached new output layer for multi-label classification and trained it using the entire training dataset of VOC 2012 Everingham et al.. We set the number of epochs to 200 and mini-batch size to 16 for all experiments. We used SGD optimizer with 0.1 of the initial learning rate, 0.9 of momentum and 0.00001 of weight decay. We drop the learning rate by $\times 0.1$ at 50, 100 and 150 epochs, respectively.

Classification of iNaturalist dataset We slightly modified the official code of iNaturalist 2018 competition Van Horn et al. (2018). Using the entire training dataset, we conducted linear evaluation using an encoder. We used SGD optimizer with 0.1 of the initial learning rate, 0.9 of momentum

and 0.00001 of weight decay. We trained an output layer for 30 epochs and drop the learning rate by $\times 0.5$ at each 10 epochs.

Semantic segmentation using VOC 2012 dataset We used PSPNet Zhao et al. (2017) as a base model for semantic segmentation. Before training, we initialized the encoder of PSPNet with an encoder trained by CL algorithm. Different with other evaluation, we trained the entire model including the encoder with the VOC 2012 dataset. For all experiments, we set 8 of mini-batch size and trained for 50000 iterations. We used SGD optimizer with 0.01 of the initial learning rate, 0.9 of momentum and 0.0001 of weight decay.