TOWARDS DIVERSE EVALUATION OF CLASS INCREMENTAL LEARNING: A REPRESENTATION LEARNING PERSPECTIVE

Sungmin Cha[1], **Jihwan Kwak**[2], **Dongsub Shim**[3], **Hyunwoo Kim**[4],
Moontae Lee[3,5], **Honglak Lee**[3], **and Taesup Moon**[2,6*]
[1]Computer Science Department at the Courant Institute of Mathematical Sciences, New York University
[2]Department of Electrical and Computer Engineering, Seoul National University
[3]Advanced Machine Learning Lab, LG AI Research
[4]Zhejiang Lab
[5]Department of Information and Decision Sciences, University of Illinois Chicago
[6]ASRI / INMC / IPAI / AIIS, Seoul National University
sungmin.cha@nyu.edu, {kkwakzi, tsmoon}@snu.ac.kr, hwkim@zhejianglab.com,
{dongsub.shim, moontae.lee, honglak.lee}@lgresearch.ai

ABSTRACT

Class incremental learning (CIL) algorithms aim to continually learn new object classes from incrementally arriving data while not forgetting past learned classes. The common evaluation protocol for CIL algorithms is to measure the average test accuracy across all classes learned so far — however, we argue that solely focusing on maximizing the test accuracy may not necessarily lead to developing a CIL algorithm that also continually learns and updates the representations, which may be transferred to the downstream tasks. To that end, we experimentally analyze neural network models trained by CIL algorithms using various evaluation protocols in representation learning and propose new analysis methods. Our experiments show that most state-of-the-art algorithms prioritize high stability and do not significantly change the learned representation, and sometimes even learn a representation of lower quality than a naive baseline. However, we observe that these algorithms can still achieve high test accuracy because they enable a model to learn a classifier that closely resembles an estimated linear classifier trained for linear probing. Furthermore, the base model learned in the first task, which involves single-task learning, exhibits varying levels of representation quality across different algorithms, and this variance impacts the final performance of CIL algorithms. Therefore, we suggest that the representation-level evaluation should be considered as an additional recipe for more diverse evaluation for CIL algorithms.

1 INTRODUCTION

Neural networks have achieved great success in various fields such as computer vision, natural language processing, and reinforcement learning (LeCun et al., 2015; Bengio et al., 2021). Among them, image classification is the first representative task leading to the significant progress of neural networks (Deng et al., 2009; Krizhevsky et al., 2009; Kingma & Ba, 2014). Furthermore, the ImageNet (Deng et al., 2009) pretrained model has been widely used as an initial model for transfer learning in other downstream tasks, such as object detection, semantic segmentation, and other classification datasets. In terms of representation learning, experimental analyses have shown that models achieving better classification accuracy learn better quality of representations, leading to better ability for transfer learning to various downstream tasks (Kornblith et al., 2019b).

However, neural networks exhibit a substantial gap with humans in their ability to continually learn from a series of tasks. To narrow this gap, research on continual learning (CL) has started, starting with image classification as the primary task (Parisi et al., 2019; Delange et al., 2021; Masana et al., 2020). Among the three scenarios of CL, Class Incremental Learning (CIL) is considered as an important sub-category that has garnered significant attention (Van de Ven & Tolias, 2019; Masana et al., 2020). This scenario models a practical scenario that can be encountered in many real-world applications and is considered as the hardest compared to the other scenarios (Van de Ven & Tolias, 2019), as the task-id is not available at inference time. In this CIL, the goal of a learning agent is to successfully integrate knowledge gained from new object classes (plasticity) from incrementally arriving data while overcoming catastrophic

* Corresponding author

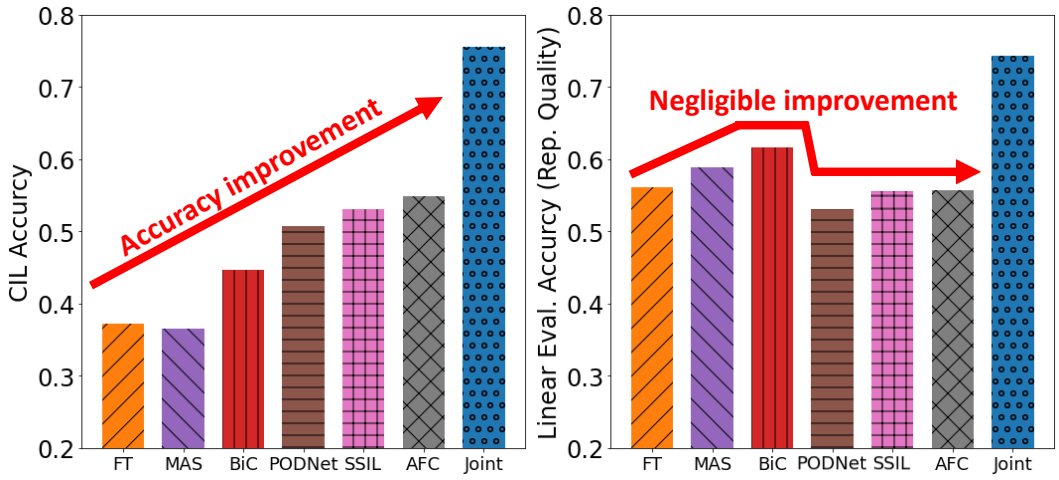


Figure 1: Experimental results of CIL using the ImageNet-100 dataset for a 10-tasks scenario. The accuracy of state-of-the-art regularization-based CIL algorithms have been gradually increasing, approaching that of Joint training (left). However, we experimentally confirm that the improvement of the quality of representations learned by them is negligible or even worse than naive baselines (right).

forgetting for knowledge of the past learned classes (stability). However, achieving this goal is difficult due to neural networks suffer from a trade-off between stability and plasticity (Mermillod et al., 2013).

The effectiveness of CIL algorithms is 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 plasticity (for learning new classes) and stability (for not forgetting past classes). Recently proposed CIL algorithms have aimed to increase the average test accuracy after learning the final task (Masana et al., 2020). As presented in Figure 1 (left), the regularization-based methods using the exemplar memory have achieved the sound progress in terms of test accuracy improvement, even approaching the performance of the model jointly trained with the entire training dataset (Wu et al., 2019; Ahn et al., 2021; Hou et al., 2019; Douillard et al., 2020; Kang et al., 2022). Similar to single-task training (*e.g.*, ImageNet training), high test accuracy of a trained model is regarded as an indicator of a better model in CIL. However, the evaluation of the representations learned by state-of-the-art CIL algorithms has not been widely discussed so far. Therefore, it remains unclear whether their performance gain comes from continually learning better representations or from other factors.

In this paper, we argue that the horse race toward simply maximizing the average test accuracy has limitations and may not necessarily lead to the development of effective CIL algorithms. Therefore, we raise the necessity to evaluate the quality of learned representations to diversify the evaluation of CIL. Our motivation comes from the experimentally confirmed relationship between the quality of representations and the classification accuracy of the classification model (Kornblith et al., 2019b). Unlike Davari et al. (2022), which analyzed forgetting of representations in continual learning using a naive baseline such as finetuning, we evaluate and analyze the representations learned by state-of-the-art regularization-based CIL algorithms. To evaluate learned representations by CIL algorithms, first, we borrow two 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 and check the test accuracy, and 2) perform transfer learning with the incrementally learned encoder to downstream tasks and report the test accuracy on those tasks. Second, to check the level of changes of the representations, we report the level of changes of representations via CKA (Centered Kernel Alignment) measure (Kornblith et al., 2019a). Additionally, we devise a metric to evaluate how closely the learned output layer weights of each CIL algorithm resemble those of a linear classifier trained for linear probing. By testing with above evaluation and analysis protocol on class incrementally learning ImageNet-100 in two major CIL scenarios (*e.g.*, 10 and 11 tasks), we obtain the following findings:

- First, despite achieving high test accuracy, state-of-the-art regularization-based CIL algorithms end up learning representations that are either inferior or comparable to those of other baselines that achieve lower test accuracy.
- Second, the majority of state-of-the-art regularization-based algorithms prioritize stability, leading to minimal enhancement in representation during CIL. Additionally, we confirm that their superior performance may stem from learning an output layer that closely resembles that of a linear classifier trained using linear probing.

- Third, the representation quality of the first task model, which is single-task learning and not heavily influenced by CIL algorithms, can vary among algorithms and significantly impact their final performance.

2 Related Work

Supervised class-incremental learning Continual learning (CL) methods can be classified into three types (Delange et al., 2021): dynamic architecture-based approaches, regularization-based methods, and exemplar-based methods. Dynamic architecture-based approaches extend the capacity of neural networks dynamically to learn a new task without catastrophic forgetting (Rusu et al., 2016; Deng et al., 2009; Mallya & Lazebnik, 2018; Schwarz et al., 2018; Hung et al., 2019; Lee et al., 2020). Regularization-based methods maintain important weights for previous tasks during training to prevent catastrophic forgetting, showing superior performance, especially for task-incremental learning (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). However, they exhibit degraded performance for class-incremental learning (CIL) (Van de Ven & Tolias, 2019). Exemplar-based methods store a subset of previous task data as exemplars and retrieve them when training a new task, showing superior performance in most CIL scenarios (Rebuffi et al., 2017; Castro et al., 2018; Wu et al., 2019; Prabhu et al., 2020). They have shown even better performance when combined with distillation-based methods (Douillard et al., 2020; Hou et al., 2019; Ahn et al., 2021; Kang et al., 2022; Asadi et al., 2023) as well as a contrastive learning-based method (Cha et al., 2021a). Additionally, some works pointed out the problem of normalization layers in CIL and proposed a novel normalization layer designed for CIL (Pham et al., 2022; Cha et al., 2023). Recently, CIL with pretrained models has gained attention, leveraging the superior representations of these models for CIL without relying on exemplar memory (Panos et al., 2023; Zhang et al., 2023; Wang et al., 2022).

Analysis of learned representations by CIL Studies probing the quality of learned representations in CIL have yielded insightful observations. Yu et al. (2020) empirically demonstrates that CIL with the encoder alone (using metric learning-based finetuning) exhibited limited forgetting, while combining the encoder and output layer (via cross-entropy-based finetuning) led to more substantial forgetting across the network. Vogelstein et al. (2020) highlights the limitation associated with accuracy-based evaluation in lifelong learning and introduces several metrics that provide a more comprehensive assessment. In the realm of task-incremental learning (TIL), Davari et al. (2022) embarks on an in-depth exploration of representation quality. Notably, they confirm that representation forgetting under naive finetuning is less pronounced than the corresponding accuracy drop in TIL. Furthermore, they demonstrate that contrastive learning-based loss functions exhibit enhanced resilience against representation forgetting, aligning with insights from continual self-supervised learning (Fini et al., 2022; Madaan et al., 2022).

In contrast to previous studies, our study distinguishes itself in two key aspects. First, we perform extensive experiments to assess the learned representations using state-of-the-art regularization-based CIL algorithms. Second, drawing from our experimental results, we highlight a common drawback in current CIL research, which tends to concentrate solely on maximizing classification accuracy. In light of these findings, we advocate for diversified evaluation methods to more effectively evaluate the quality of representations learned by CIL algorithms.

3 Towards Diverse Evaluation of CIL from a Representation Perspective

3.1 Problem formulation and preliminaries

In this section, we briefly introduce the preliminaries and problem formulation of our paper. We follow the general settings and problem formulation of class-incremental learning (CIL) proposed in previous papers (Rebuffi et al., 2017; Van de Ven & Tolias, 2019; Masana et al., 2020).

Notations and settings We assume a sequential task setting, where $t \in \{1, \dots, T\}$ represents the t^{th} task. Task-specific training and test datasets at task t are denoted as D_t^{tr} and D_t^{te}, respectively. 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. The target label y_t is assumed to be sampled from a task-specific class set $\mathcal{C}_t$ which are disjoint across different tasks, *i.e.* $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset \; \forall i \neq j \in \{1, \dots, T\}$. Exemplar memory is allocated to store and replay a small number of data instances of previous tasks. More specifically, exemplar memory which holds data seen until task $t-1$ is denoted as $\mathcal{M}_{t-1}$ and is used for training at task t. In this paper, we consider a class-balanced memory which is simple in that it stores equal number of images per class and is known to be efficient (Prabhu et al., 2020; Castro et al., 2018; Hou et al., 2019; Douillard et al., 2020).

At task t, a classification model $f_{\boldsymbol{\theta}_t} = (g_{\boldsymbol{\psi}_t} \circ h_{\boldsymbol{\phi}_t})$ is trained, where $h_{\boldsymbol{\psi}_t}$ and $g_{\boldsymbol{\phi}_t}$ indicates the encoder and the output layer of the model, respectively. In this paper, we consider and compare CIL algorithms that use the cross-entropy

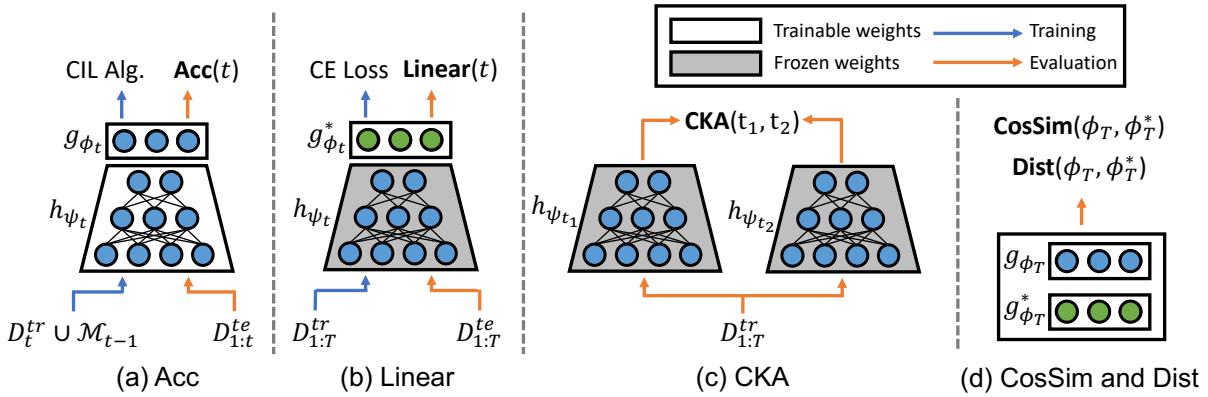


Figure 2: This figure illustrates proposed and used evaluation methods in this paper. (a): the standard evaluation method that measures classification accuracy on test data following training task t. (b): linear probing evaluation for a model trained on task t. (c): measuring CKA between representations of two models trained on different tasks. (d): a comparison of weights in the output layer.

(CE) objective function as a main training objective. Note that the model trained with the entire training datasets until task t is denoted as *joint* which is an oracle case. $f_{\boldsymbol{\theta}_t}$ is trained on $D_t^{\text{tr}} \cup \mathcal{M}_{t-1}$ for multiple epochs (offline training) at task t, and evaluated on $D_{1:t}^{\text{te}}$. In CIL, task ID, an additional supervisory signal, is not provided as it adopts a shared output layer.

Conventional metrics of CIL Among various ordinary metrics for CIL, we adopt two general metrics for evaluating the performance of CIL algorithm (Masana et al., 2020): **Acc**(t) (shown in Figure 2) and **AvgAcc**(t). **Acc**(t) is the test accuracy of $f_{\boldsymbol{\theta}_t}$ on $D_{1:t}^{\text{te}}$, and **AvgAcc**(t) is the average of **Acc**(t) from the first task to the t-th task, *i.e.,* **AvgAcc**$(t) = \frac{1}{t}\sum_{l=1}^{t}$**Acc**$(l)$.

3.2 Proposed Evaluation Method for Analysis from a Representation Perspective

To comprehensively assess the improvements in learned representations by CIL algorithms, we deploy a structured evaluation framework encompassing both in-domain and out-domain perspectives.

(1) **In-domain evaluation: Linear**(t) **and** k**-NN**(t) To compare the improvement of representations learned by CIL algorithms, we borrow the evaluation methods used in representation learning research (Zbontar et al., 2021; He et al., 2020): Linear evaluation and k-NN classification. As shown in Figure 2 (b), representations of each encoder $h_{\boldsymbol{\psi}_t}$ is evaluated by freezing the encoder $h_{\boldsymbol{\psi}_t}$ and conducting a linear probing by re-training the final linear layer to obtain an estimated linear classifier $g_{\boldsymbol{\phi}}^*$. k-NN classifier ($k = 20$) is also constructed with the frozen encoder. Note that the entire training dataset of a given CIL scenario, $D_{1:T}^{\text{tr}}$, is used to train the estimated linear classifier or to formulate k-NN classifier and that the is entire test dataset $D_{1:T}^{\text{te}}$ is used for evaluation.

(2) **Out-domain evaluation: CLS**(t) To further evaluate the quality of the learned representations in more general aspects, we conduct experiments of transfer learning with out-domain datasets as well. We consider three downstream tasks of classification, namely STL-10 (Coates et al., 2011), CUB200 (Wah et al., 2011), and resized (96×96) 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.

(3) **Representation similarity comparison: CKA**(t_1, t_2) We compare the degree of changes of learned representations during a task change in CIL from t_1 to t_2 by measuring their similarity using CKA (Kornblith et al., 2019a). That is, as shown in Figure 2 (c), we measure CKA between $h_{\boldsymbol{\psi}_{t_1}}$ and $h_{\boldsymbol{\psi}_{t_2}}$ by using entire training dataset $D_{1:T}^{\text{tr}}$.

(4) **Comparison of weights of output layer** For further analysis for output layer, we propose to conduct comparison between a classifier layer $g_{\boldsymbol{\phi}_T}$ trained during CIL and an estimated linear classifier $g_{\boldsymbol{\phi}_T}^*$ trained for linear probing (Davari et al., 2022), for the final T-th task's model. Let $W \in \mathbb{R}^{D\times C}$ and $W^* \in \mathbb{R}^{D\times C}$ are parameters of the classifier layer of $g_{\boldsymbol{\phi}_T}$ and $g_{\boldsymbol{\phi}_T}^*$, respectively. D denotes the dimension of an output feature of $h_{\boldsymbol{\psi}_T}$ and C stands for the number of the whole classes. To compare both parameters, we calculate both cosine similarity and L_2 distance

between them as below:

$$\textbf{CosSim}(\phi_T, \phi_T^*) = \frac{1}{C}\sum_{i}^{C}\frac{(W_i)^\intercal W_i^*}{\|W_i\|_2\|W_i^*\|_2} \qquad \textbf{Dist}(\phi_T, \phi_T^*) = \frac{1}{C}\sum_{i}^{C}\|W_i - W_i^*\|_2$$

where i denotes an index of column axis and $W_i, W_i^* \in \mathbb{R}^D$. Note that, when $\textbf{CosSim}(\phi_T, \phi_T^*)$ is high and $\textbf{Dist}(\phi_T, \phi_T^*)$ is low at the same time, it means the classifier layer is close to the estimated linear classifier.

4 Experimental Setup

Baselines As discussed in the Related Work section, class incremental learning (CIL) research has evolved in various forms, such as regularization-based, exemplar-based, and model expansion-based methods. Recently, approaches using contrastive learning and pretrained models have also been proposed. However, in this paper, we focus on analyzing and evaluating regularization-based methods that leverage exemplar memory. The rationale for this focus is twofold: 1) scontrastive learning-based method have already demonstrated gradual improvements in representation (Cha et al., 2021a; Fini et al., 2022) 2) CIL with a pretrained model starts with superior representations, leading to either freezing these models or aiming for minimal changes, which sets them apart from previous studies (Panos et al., 2023; Zhang et al., 2023; Wang et al., 2022). On the other hand, while regularization-based methods using the exemplar memory have been studied for a long time and continue to report excellent results, the analysis and experimentation regarding the learned representations of these methods are generally overlooked. The baseline algorithms used in our experiments and brief descriptions of them are as follows:

1) Finetuning (FT): It is a naive approach using fine-tuning a model with exemplars.
2) MAS (Aljundi et al., 2018): It measures the importance of each weight that constitutes the model using gradients and uses this importance as the strength of regularization to overcome catastrophic forgetting.
3) BiC (Wu et al., 2019): It overcomes catastrophic forgetting by performing knowledge distillation from the previously learned model, as in LWF (Li & Hoiem, 2017). Additionally, biased prediction issues in the output layer are resolved through post-processing on the prediction score. We also report results of BiC (w/o BC) which indicates results without the bias correction post-processing for output logits.
4) PODNet (Douillard et al., 2020): It uses a more sophisticated spatial-based distillation loss to balance between learning new classes and forgetting previously learned classes.
5) SSIL (Ahn et al., 2021): It uses separated softmax and task-wise knowledge distillation to alleviate biased predictions.
6) AFC (Kang et al., 2022): It calculates the importance in each feature map and proposes regularization using this importance to learn new knowledge well while preserving previously learned knowledge.

For our experiments, we train models using the official codes of PODNet, SSIL, and AFC for each algorithm. For BiC, we conduct experiments using the implementation in the official code of PODNet, and for FT and MAS, we conduct experiments using the CIL framework proposed in (Masana et al., 2020).

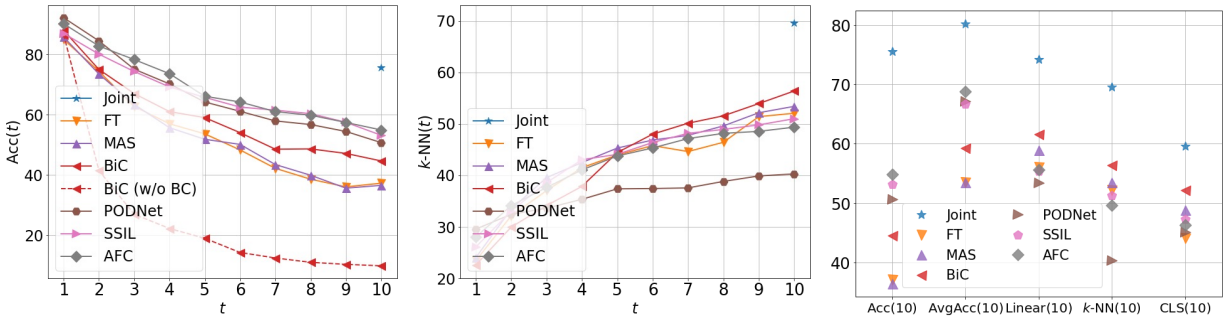
CIL scenarios We consider two CIL scenarios using the ImageNet-100 dataset (Deng et al., 2009). The first scenario, denoted as **10-tasks**, is consisting of 10 tasks each with 10 classes that are continuously learned. The second scenario, denoted as **11-tasks**, is a scenario where 50 classes are learned in the base task (first task), followed by 10 continuous tasks each with 5 classes.

Other settings For all experiments, we use the ResNet-18 (He et al., 2016). All the baseline models are trained with the same hyperparameters proposed by each algorithm. For additional training to get a result of $\textbf{Linear}(t)$, we train a new output layer with a mini-batch size of 256 with 30 epochs for in-domain dataset and 100 epochs for out-domain datasets. We use SGD optimizer with an initial learning rate of 0.1 and momentum of 0.9 and decay rate of 0.0001. We apply a schedule that multiplies the learning rate by 0.1 at $\{10, 20\}$ and $\{40, 80\}$ epochs for in-domain and out-domain evaluation, respectively. We conduct experiments for three seeds and report averaged results of them.

Note that, in the Appendix, we present the experimental setting and experimental results for alternative CIL algorithms, such as LUCIR, as well as for a distinct dataset, such as CIFAR-100.

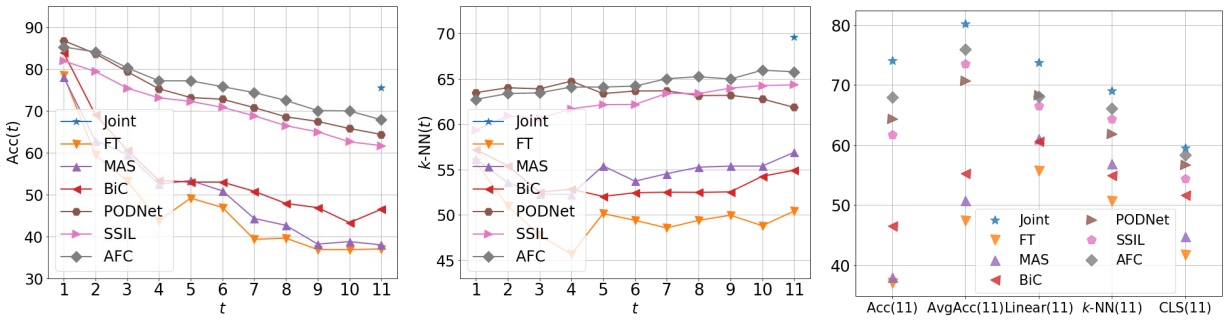
5 Experimental Results with the Proposed Evaluation

We then evaluate several regularization-based class incremental learning (CIL) algorithms using the proposed evaluation method. Our findings are summarized into three key points, and we present the experimental results sequentially.



(a) Acc(t). (b) k-**NN**(t) (c) All eval. metrics

Figure 3: The experimental results of regularization-based CIL algorithms for a 10-task scenario using the ImageNet-100 dataset. "Joint" refers to the performance of the upper bound case using the entire datasets.



(a) Acc(t). (b) k-NN(t). (c) All eval. metrics.

Figure 4: The experimental results of regularization-based CIL algorithms for a 11-task scenario using the ImageNet-100 dataset. "Joint" refers to the performance of the upper bound case using the entire datasets.

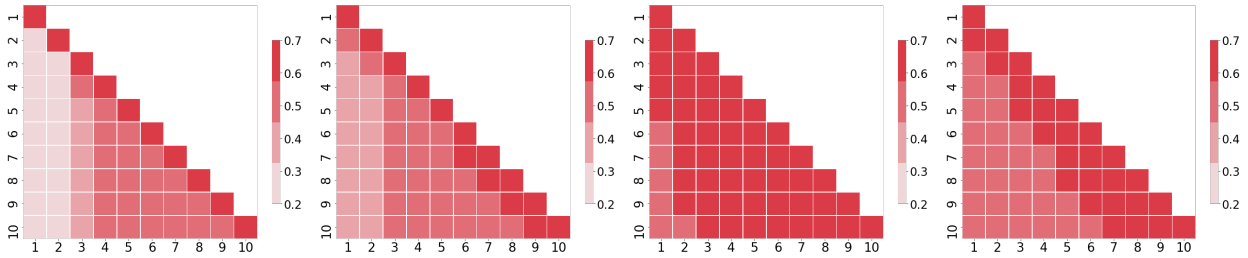
5.1 Achieving superior performance in conventional metrics does not always mean learning a superior representation

For the ImageNet trained models, experimental evidence indicates that models achieving superior classification accuracy also learn a superior representation (Kornblith et al., 2019b). On the other hand, many regularization-based CIL methods devise novel regularization motivated by the goal of learning new knowledge (*i.e.*, representation) effectively while retaining knowledge from previous tasks. As a result, they demonstrate superior performance in the final classification accuracy (Masana et al., 2020). Inspired by these findings, we ask the following question: Do regularization-based CIL algorithms using exemplar memory attain their excellence by learning a superior representation? To explore this, we construct experiments on 10-tasks and 11-tasks scenarios using the ImageNet-100 dataset and apply the proposed analysis after training models using selected regularization-based CIL algorithms.

Figure 3 presents the experimental results for the 10-tasks scenario. First, Figure 3(a) shows CIL algorithm's **Acc**(t). As reported in their paper, state-of-the-art algorithms (*i.e.*, AFC, SSIL, and PODNet) demonstrate the most superior performance, with BiC following closely behind. Additionally, not only MAS and BiC (w/o BC) achieve worse performance compared to them but also BiC (w/o BC) shows degraded performance as reported in their paper (Wu et al., 2019). Figure 3(b) and 3(c) show the result of applying our proposed evaluation. From these results, we observe several interesting findings: First, from the k-**NN**(t) results in Figure 3(b), we can confirm that the results of conventional metrics do not always align with the quality of learned representations. For instance, in k-**NN**(t), BiC and MAS show superior representations compared to the state-of-the-art algorithms. Specifically, considering that the representation of BiC is the same as BiC (w/o BC) (Wu et al., 2019), CIL using only the knowledge distillation method (*e.g.*, LWF) achieves significantly degraded performance in **Acc**(t) due to biased prediction but demonstrates a better representation learning capability. Furthermore, the additional results in Figure 3(c) highlight these trends more distinctly. Despite the state-of-the-art algorithms achieving superior performance in the conventional metrics (*i.e.*, **Acc(10)** and **AvgAcc(10)**), they exhibit the same trend in all metrics evaluating representation quality as before. Particularly, despite PODNet achieving relatively superior performance in the conventional metrics, the representation they learned are significantly worse than others.

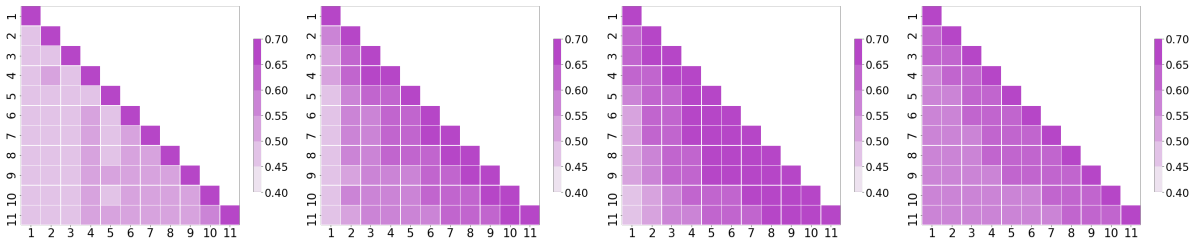
In contrast, the results of the 11-tasks in Figure 4 exhibit a different trend. First, from the **Acc**(t) in Figure 4(a), we can observe that the state-of-the-art algorithms continue to demonstrate the most superior performance and are almost approaching performance of Joint. Unlike the 10-tasks scenario, the results in Figure 4(b) demonstrate that these state-of-the-art methods learn better representations than others. Particularly, AFC and SSIL show a slight improvement in quality of learned representations over tasks. Similarly, Figure 4(c) demonstrate the state-of-the-art algorithms achieve not only superior performance in the conventional metrics but also superior representation learning.

We have observed that several state-of-the-art CIL algorithms exhibit completely different trends in representation learning between the 10-tasks and 11-tasks scenarios. The only difference between these scenarios is that in the 11-tasks scenario, the models start by learning half of the classes from the first task. Taking this difference into consideration, we conduct additional analysis to understand the reasons behind these results and to gain further insights into characteristics of these algorithms.



(a) Joint (b) BiC (c) PODNet (d) AFC

Figure 5: CKA(t_1, t_2) in 10-tasks scenario for $t_1, t_2 \in \{1, \ldots, 10\}$. Each $CKA(t_1, t_2)$ quantifies the similarity between representations of two models trained on distinct tasks. A deep red color indicates a higher level of similarity compared to a lighter shade of red.



(a) Joint (b) BiC (c) PODNet (d) AFC

Figure 6: CKA(t_1, t_2) in 11-tasks scenario for $t_1, t_2 \in \{1, \ldots, 11\}$. Each $CKA(t_1, t_2)$ quantifies the similarity between representations of two models trained on distinct tasks. A deep purple color indicates a higher level of similarity compared to a lighter shade of purple.

5.2 Most regularization-based CIL algorithms significantly prioritize stability

In previous experimental results, we observed that state-of-the-art algorithm learn inferior representations in the 10-tasks scenario. However, in the 11-tasks scenario, while they learn superior representations, there is no significant difference when compared to the representations learned in the first task. This led us to hypothesize that the poor sequential representation updates are due to the strong stability of these algorithms. Consequently, we conduct additional analysis using CKA to investigate further.

Figure 5 and 6 shows **CKA**(t_1, t_2) of Joint, BiC, PODNet, and AFC, which compares the representation similarity between $h_{\psi_{t_1}}$ and $h_{\psi_{t_2}}$. For example, **CKA**$(5, 1)$ and **CKA**$(5, 3)$ of Joint in Figure 5(a) shows that an encoder at task 5 has less representation similarity to the encoder at task 1 compared to the encoder at task 3. From the results of both the 10-tasks and 11-tasks scenarios, we can draw the following findings: First, the Joint exhibits a high similarity in representations between adjacent tasks, but ultimately, it undergoes progressive changes throughout the CIL process, resulting in superior learned representations at the final task. However, the representation similarity of PODNet and AFC remains relatively high. This indicates that these algorithms place a heavy emphasis on stability, leading to minimal changes in representations.

This aligns with the k-**NN** results in the previous section. In the 11-task scenario, the superior representation learned in the first task is consistently maintained, enabling these algorithms to achieve better results by the final task compared to other algorithms. Conversely, in the 10-tasks scenario, the representation learned in the first task is not as superior, preventing the algorithms from learning improved representations during the CIL process, which results in poorer performance compared to other baselines.

5.3 The superior performance of state-of-the-art algorithms might be attributed to their ability to learn a good output layer.

Additionally, one remaining question is how does the state-of-the-art algorithms (*i.e.*, PODNet, SSIL, and AFC) can still achieve high **Acc**(t) and **AvgAcc**(t) even with *poor representations*? To obtain an answer to this question, we conduct an analysis on weights of output layer. Figure 7 compares the similarity of weights between an original classifier learned in last task (*i.e.*, g_{ϕ_T}) and the estimated linear classifier trained for linear probing (*i.e.*, $g^*_{\phi_T}$). Note that results for PODNet and AFC are absent, as these algorithms adopt specialized cosine classifiers to address biased prediction in output layer. First, Joint demonstrates both the highest cosine similarity (**CosSim**($\phi_T, \phi T$)) and lowest L_2 distance (**Dist**($\phi T, \phi_T$)). Conversely, BiC(w/o BC) exhibits the lowest similarity to the estimated linear classifier. Furthermore, SSIL achieves a higher similarity to the estimated linear classifier compared to FT and MAS, benefiting from its novel approach which alleviates the biased prediction. Note that the order of similarity of output layers mirrors the performance order in Figure 3(a). Considering this trend alongside the experimental results on representation quality in Figure 3(c), it is evident that SSIL learns representations comparable to those of FT and MAS, and inferior to BiC. Despite this, SSIL achieves overwhelmingly superior performance in conventional metrics, significantly benefiting from learning an effective output layer rather than from learning superior representations.

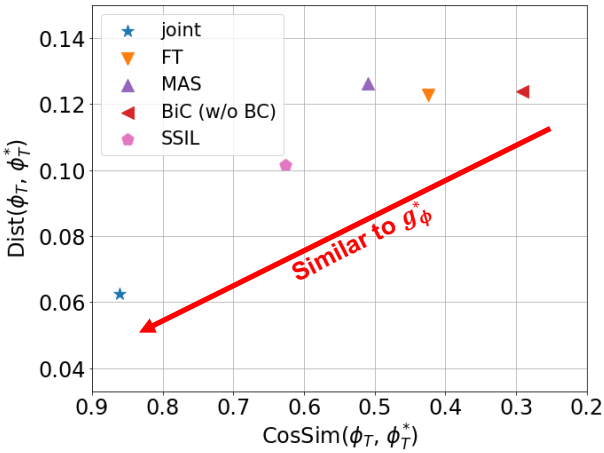


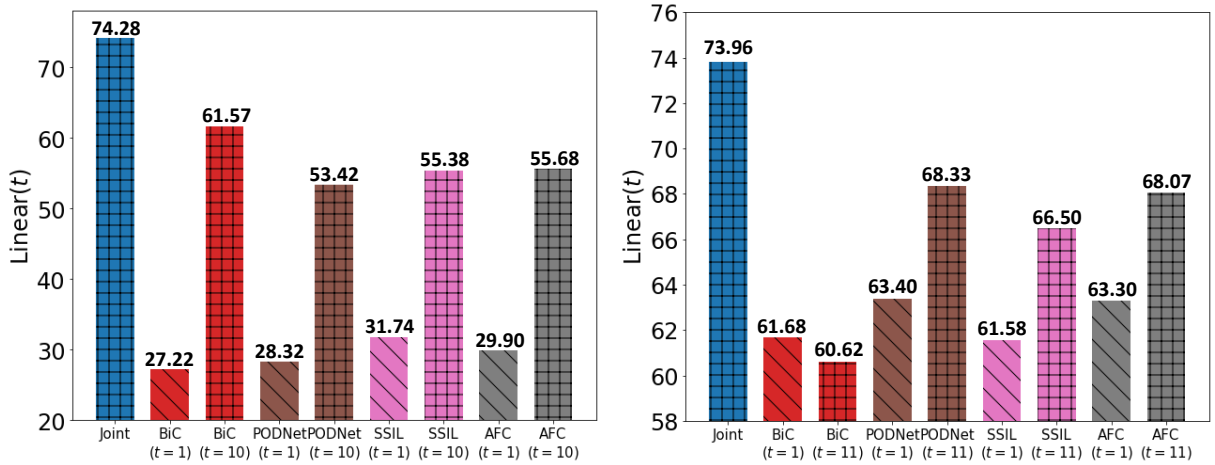
Figure 7: Experimental results of CosSim(ϕ_T, ϕ^*_T) and Dist(ϕ_T, ϕ^*_T). We use a model trained by each algorithm in the 10-tasks scenario.

Similarly, we can draw a comparable inference for other state-of-the-art algorithms such as PODNet and AFC. Even though these algorithms learn representations that are inferior or not significantly different from those of other baselines, they can still achieve superior performance in conventional metrics thanks to the use of a specialized cosine classifier. This indicates that their performance advantage likely stems not from their novel regularization devised for controlling a trade-off between stability and plasticity in representations, but rather from the sophisticated design of the specialized output layer they commonly employ.

5.4 The quality of the representation learned in the first task can have a significant impact on the final evaluation

In the previous section, we noted significant differences in the evaluation results of the first task models among various algorithms, as illustrated in Figures 3(b) and 4(b) (e.g., k-**NN**(1)). We recognize that learning the initial task in CIL resembles single-task learning and is mostly unaffected by a particular regularization-based CIL algorithm. Considering the focus of these CIL algorithms on stability, we hypothesize that the disparities among these first task models could significantly impact overall performance. To validate this hypothesis, we conduct additional experiments.

We compare the linear probing results of the first task model trained by each algorithm in the 10-tasks and 11-tasks scenarios, as shown in Figure 8. From this figure, despite employing the same dataset in the same scenario, we again observe that the linear probing results of the first task model exhibit variance across algorithms, as evidenced by the distinct **Linear**(1) outcomes in the figure. We specifically notice a consistent pattern in the performance discrepancies and ranking between **Linear**(1) and **Linear**(T) across all algorithms, as observed in the results of PODNet, SSIL, and AFC. For instance, in the 11-task scenario, the performance gap between PODNet and SSIL in **Linear**(1) is approximately 2%, mirroring their difference in **Linear**(11). This observation reinforces the notion that variations in



(a) 10-tasks (b) 11-tasks with the base task

Figure 8: Experimental results of linear probing for each algorithm.

learning the first task can indeed affect the final performance assessment, leading us to conduct experiments focused on aligning the first task model as closely as possible.

Table 1: Experimental results for the 10- and 11-tasks scenarios.

Alg.	10-tasks						11-tasks					
	Conv. Metrics		In-domain			Out-domain	Conv. Metrics		In-domain			Out-domain
	Acc(10)	Avg(10)	**Linear**(1)	Linear(10)	k-NN(10)	CLS(10)	Acc(11)	Avg(11)	**Linear**(1)	Linear(11)	k-NN(11)	CLS(11)
Joint	75.56	80.23	-	74.28	69.62	59.57	75.40	84.23	-	73.96	74.42	59.80
FT	37.20	53.48	**33.12**	56.08	52.04	44.09	37.01	47.35	**61.22**	56.12	56.12	41.72
MAS	36.48	53.55	**32.66**	58.94	53.38	48.91	38.00	50.85	**60.62**	61.08	61.08	44.79
BiC	44.60	59.28	**27.22**	61.58	56.36	52.20	46.50	55.31	**61.67**	60.61	63.40	51.71
PODNet	50.70	66.70	**28.32**	53.42	40.26	45.09	64.40	73.48	**63.40**	68.33	61.86	56.67
SSIL	53.14	67.12	**31.74**	55.38	50.98	47.28	61.72	70.41	**61.58**	66.50	61.48	54.39
AFC	54.90	68.83	**29.90**	55.68	49.33	46.39	67.90	75.89	**63.30**	68.07	63.30	58.38

Table 1 presents the results of conventional metrics and all proposed evaluations obtained from the 10-tasks and 11-tasks and we again confirm that **Linear**(1) of each algorithm is different. Our objective is to equalize the performance of **Linear**(1). For this, we set the same first task model with the lowest performance in each scenario for all algorithms. For example, in the case of 10-tasks, we train a first task model with different epochs to obtain a model whose **Linear**(1) result closely matches 28.32 (the **Linear**(1) result of PODNet). Once we obtain a first task model with the similar result, we proceed with CIL using the original hyperparameters. The experimental results for this approach are shown in Table 2. Instances marked with an asterisk (*) indicate results obtained using the unified first task model. Experimental results in the figure show that employing the unified first task model led to a decrease in performance across both scenarios. In the 10-tasks scenario, SSIL* and AFC* show performance decreases of 1-2% in **Acc**(10) compared to their original results. Notably, for SSIL*, the difference in **Acc**(10) compared to PODNet reduces from roughly 4% to around 0.6%. Similarly, in the 11-tasks scenario, PODNet* and AFC* suffer from a 2-4% decrease compared to the original results. As a result, although PODNet initially exhibits about a 3% higher **Acc**(11) than SSIL, PODNet* ultimately performs worse than SSIL in terms of **Acc**(11).

Table 2: Experimental results with a unified first task model. * denotes using the unified first task model.

Alg.	10-tasks					
	Acc(T)	Avg(10)	**Linear**(1)	Linear(10)	k-NN(10)	CLS(10)
SSIL*	51.34	65.74	**28.48**	54.54	52.26	45.28
AFC*	53.70	67.85	**28.48**	51.02	46.82	43.28
Alg.	11-tasks					
	Acc(11)	Avg(11)	**Linear**(1)	Linear(11)	k-NN(11)	CLS(11)
PODNet*	60.90	70.11	**61.28**	63.28	55.26	51.67
AFC*	65.90	74.48	**61.24**	64.34	61.70	56.80

Additionally, comparing Table 1 and Table 2 highlights shifts in representation quality when we establish the unified first task model. This is particularly noticeable in the 11-tasks scenario, where adjustments in the first task model

significantly affect the representation quality of the final model. For instance, both PODNet and AFC begin learning from a model with approximately a 2% decrease in **Linear**(1). This consequently results in notable performance declines of around 4% at **Linear**(11) and 2-6% at k**-NN**(11) and **CLS**(11).

In this section, we propose novel findings through evaluating the representations of the first task model learned by each CIL algorithm. Note that, during the hyperparameter tuning process, most algorithms select the best hyperparameters that achieve the highest **Acc** or **AvgAcc** after learning the final task. Considering that most state-of-the-art regularization-based CIL algorithms heavily prioritize stability, it is plausible that the optimal hyperparameters, which can learn the best representation in the first task and preserve it in subsequent tasks, could be chosen as the best hyperparameters, especially in scenarios where many classes are learned in the first task (*e.g.*, 11-tasks). Indeed, when we equalized the first task model, we observed a significant reduction in the differences in final performance across algorithms. This suggests the importance of evaluating the quality of model representations in each task to accurately assess the performance gains of each CIL algorithm.

6 Concluding Remarks, Limitation and Future Work

Key insights from experiments Based on experiments with state-of-the-art regularization-based CIL algorithms, we first confirm that evaluation results based on conventional metrics not align with the evaluation results of representations learned by each algorithm. Second, due to the heavy emphasis on stability in most state-of-the-art regularization-based CIL algorithms, representations do not change significantly during the CIL process. As a result, these algorithms achieve significant advantages primarily in scenarios where the first task involves learning many classes. Third, we demonstrate that the performance gains of these algorithms may stem more from a sophisticated output layer than from novel regularization terms. Lastly, we note that the representations learned in the first task can vary significantly across algorithms, significantly impacting the final evaluation.

The importance of evaluation in a representation learning perspective Neglecting the assessment of representations can lead to favorably evaluating algorithms that learn a good output layer despite having poorer representation quality. This not only inaccurately evaluates CIL algorithms but also limits their maximum potential to subpar representations instead of achieving the representation of the joint model. Building on these findings, We question whether the prevailing evaluation method, focused solely on classification accuracy, truly captures the factors that drive the performance improvement of each algorithm. In this regard, we highlight the need for diverse forms of evaluation, especially from a representation learning perspective, to precisely understand the performance gain of them.

Limitation and future work One limitation of our study is that we solely focus on regularization-based CIL algorithms. Given the recent advancements in model expansion-based methods and the prevalence of CIL studies utilizing pretrained models, it may be necessary to conduct similarly diverse evaluations in these areas as well. We leave this as a consideration for future work.

Acknowledgements

This work was supported in part by the National Research Foundation of Korea (NRF) grant [No.2021R1A2C2007884] and by Institute of Information & communications Technology Planning & Evaluation (IITP) grants [RS-2021-II211343, RS-2021-II212068, RS-2022-II220113, RS-2022-II220959] funded by the Korean government (MSIT). It was also supported by SNU-NAVER Hyperscale AI Center and AOARD Grant No. FA2386-23-1-4079.

A Detailed Experimental Settings

Experimental settings of CIL algorithm We achieve the result of CIL algorithms, FT, MAS (Aljundi et al., 2018) and LWF (Li & Hoiem, 2017) by implementing the CIL framework code proposed by (Masana et al., 2020). We do not modify the default hyperparameters for each algorithm. We train these algorithms for 100 epochs for each task using the SGD optimizer with an initial learning rate of 0.1, momentum of 0.9, and weight decay of 0.0001. We also set a learning rate schedule that dropped the learning rate by a factor of 0.1 at 40 and 80 epochs, respectively. For all experiments, we use a mini-batch size of 256. We employed random sampling as the sampling algorithm for the exemplar memory.

We evaluate several regularization state-of-the-art CIL algorithms, including PODNet (Douillard et al., 2020), SSIL (Ahn et al., 2021), and AFC (Kang et al., 2022). To ensure fair comparisons, we run the official code for each algorithm without modifying not only the default hyperparameters but also other settings for training, such as learning rate, epochs, and mini-batch size. Furthermore, we obtain experimental results for LUCIR (Hou et al., 2019) and BiC (Wu et al., 2019) using the code implemented in (Douillard et al., 2020), also without any modification.

Linear probing We retrain the output layer while freezing the encoder. Specifically, we train the output layer for 30 epochs using a mini-batch size of 256, and utilize the SGD optimizer with an initial learning rate of 0.1, momentum of 0.9, and weight decay of 0.0001. We implement a learning rate schedule, which decreases the learning rate by a factor of 0.1 at the 10th and 20th epochs, respectively.

k-NN evaluation For all experiments, we utilize the k-NN implementation ($k = 20$) provided by scikit-learn Pedregosa et al. (2011). In the classification process, we first fit the k-NN with the outputs of the encoder for the given inputs, and subsequently classify the test data using the k-NN classifier.

Three downstream tasks We select CIFAR-10 (Krizhevsky et al., 2009), STL-10 (Coates et al., 2011), and CUB-200 (Wah et al., 2011) as downstream tasks for out-of-domain evaluation. For CIFAR-10, we randomly select 5,000 training images from the entire training dataset and resized the input images to 96×96 pixels. For STL-10 and CUB-200, we use the entire training dataset and maintained their original image sizes. We train only a newly added output layer while freezing the encoder. For CIFAR-10 and CUB-200, we train the output layer for 100 epochs using a mini-batch size of 128 and use SGD optimizer with an initial learning rate of 0.1, a momentum of 0.9, and a weight decay of 0.0001. We set the learning rate schedule to drop the learning rate by a factor of 0.1 at 40 and 80 epochs, respectively. In the case of STL-10, we change the number of epochs to 10 and the initial learning rate to 0.005.

Experimental settings for unifying the base task's model To unify the representation quality of the first task, we only reduce the number of epochs for the first task training of CIL algorithms that learn relatively better representations than others. The used number of training epochs for the first task is shown in Table 3.

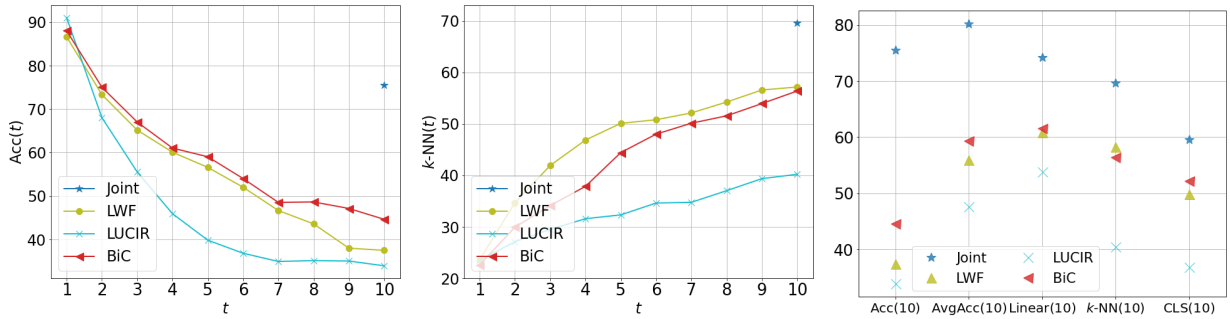
Table 3: The used numnber of training epochs for 10- and 11-tasks scenarios.

	10-tasks		11-tasks	
	SSIL	AFC	PODNet	AFC
First task Epochs	50	45	60	45

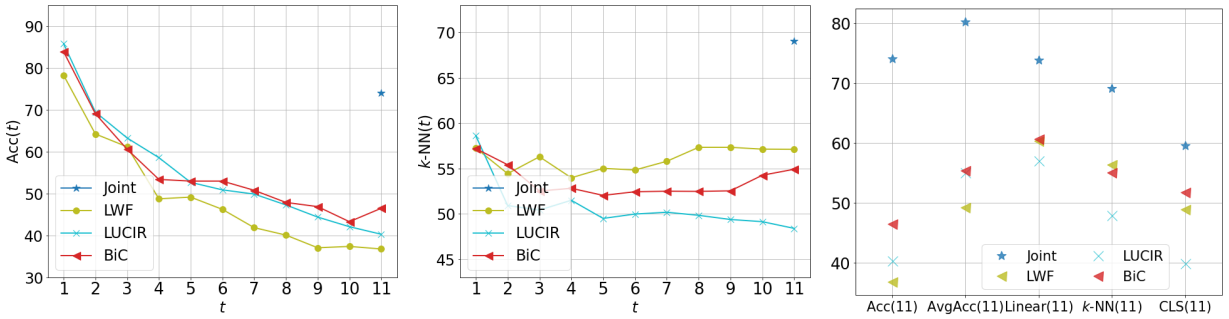
B Additional Experimental Results

B.1 Experimental analysis for other CIL algorithms

In our experiment using the ImageNet-100 dataset, we conduct additional experiments on LWF (Li & Hoiem, 2017) and LUCIR (Hou et al., 2019), and the results are shown in Figure 9. The results for BiC (Wu et al., 2019) are added for comparison and are consistent with the results in the manuscript. We are able to confirm experimental results similar to the findings in the manuscript in two scenarios (10-tasks and 11-tasks). First, since the learning of the BiC encoder is carried out through a form of knowledge distillation similar to LWF, the evaluation results for representation quality between LWF and BiC show no significant difference. Second, in the case of LUCIR, although it achieved higher **Acc**(t) and **AvgAcc**(t) than LWF in the situation of learning from the base model (11-tasks), it demonstrates that the representation quality learned by this algorithm is significantly inferior to LWF.

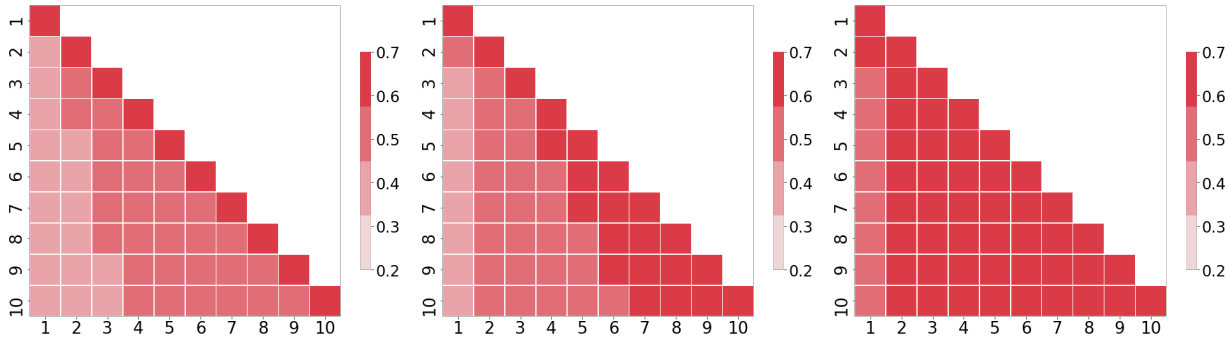


(a) 10-tasks, Acc(t). (b) 10-tasks, k-**NN**(t) (c) 10-tasks, All eval. metrics

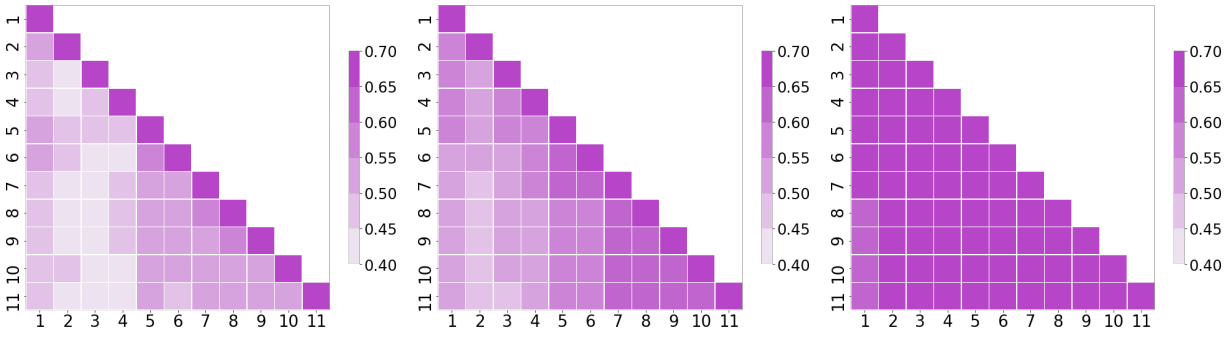


(d) 11-tasks, Acc(t). (e) 11-tasks, k-**NN**(t) (f) 11-tasks, All eval. metrics

Figure 9: Additional experimental results of LWF and LUCIR for the scenario of 10-tasks and 11-tasks.



(a) FT (10-tasks) (b) MAS (10-tasks) (c) SSIL (10-tasks)

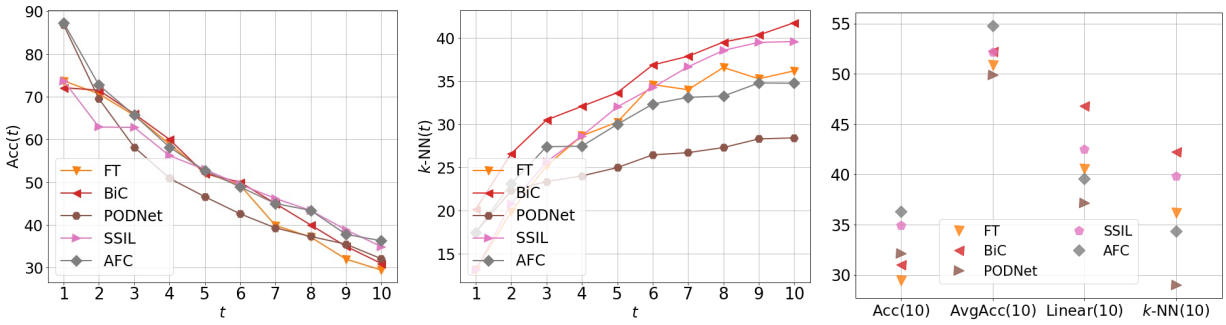


(d) FT (11-tasks) (e) MAS (11-tasks) (f) SSIL (11-tasks)

Figure 10: CKA(t_1, t_2) in 10-tasks and 11-tasks scenario.

B.2 CKA RESULTS FOR OTHER CIL ALGORITHMS

Figure 10 shows the **CKA**(t_1, t_2) results for other algorithms in the experiment using ImageNet-100. In the case of 10-tasks, first, FT and MAS show results similar to the Joint in the manuscript. Second, SSIL maintains a relatively strong similarity. Through this, we can further confirm that each SOTA algorithm focuses on stability to prevent significant changes in learned representations, leading to poorer representation learning compared to FT and MAS in the 10-tasks scenario. In the case of 11-tasks, since a relatively large amount of knowledge (half of the total class number) is learned in the first task, algorithms that focus on stability can achieve relatively superior results. Taking this into account, FT and MAS show significant changes in representation, while SSIL does not, showing a similar trend to the evaluation results for representation quality for each algorithm (See Figure 4).



(a) Acc(t). (b) k-**NN**(t) (c) All eval. metrics

Figure 11: Additional experimental results with CIFAR-100 for the scenario of 10-tasks.

B.3 EXPERIMENTAL ANALYSIS WITH CIFAR-100

To investigate whether the analysis results proposed in the manuscript are dataset-dependent, we conduct CIL experiments using CIFAR-100 for the 10-tasks scenario. We use five representative algorithms (FT, BiC, PODNet, SSIL, and AFC) and conduct experiments with their reported default hyperparameters. All experiments are conducted using the ResNet-18 model, and for SSIL, which has no experiments on CIFAR-100 in the their paper, we apply the same number of epochs (160) used for training in PODNet and AFC. Additionally, we only conduct in-domain evaluation.

Figure 11 shows the experimental results, and we obtain analysis results similar to those of ImageNet-100 in the manuscript. First, we confirm that AFC and SSIL achiev relatively superior **Acc**(t) and **AvgAcc**(t), but learned representations are inferior to those learned by BiC. Second, we observe that PODNet learns significantly worse representations compared to other algorithms on CIFAR-100.

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.

Nader Asadi, MohammadReza Davari, Sudhir Mudur, Rahaf Aljundi, and Eugene Belilovsky. Prototype-sample relation distillation: towards replay-free continual learning. In *International Conference on Machine Learning (ICML)*, pp. 1093–1106. PMLR, 2023.

Yoshua Bengio, Yann Lecun, and Geoffrey Hinton. Deep learning for ai. *Communications of the ACM*, 64(7):58–65, 2021.

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.

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 (ICLR)*, 2021b.

Sungmin Cha, Sungjun Cho, Dasol Hwang, Sunwon Hong, Moontae Lee, and Taesup Moon. Rebalancing batch normalization for exemplar-based class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20127–20136, 2023.

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.

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

MohammadReza Davari, Nader Asadi, Sudhir Mudur, Rahaf Aljundi, and Eugene Belilovsky. Probing representation forgetting in supervised and unsupervised continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16712–16721, 2022.

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.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255. Ieee, 2009.

Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *In Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 86–102. Springer, 2020.

Enrico Fini, Victor G Turrisi da Costa, Xavier Alameda-Pineda, Elisa Ricci, Karteek Alahari, and Julien Mairal. Self-supervised models are continual learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9621–9630, 2022.

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 (CVPR)*, 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 (CVPR)*, 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 (CVPR)*, 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 (NeurIPS)*, 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.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning (ICML)*, pp. 3519–3529. PMLR, 2019a.

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 (CVPR)*, pp. 2661–2671, 2019b.

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

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

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.

Divyam Madaan, Jaehong Yoon, Yuanchun Li, Yunxin Liu, and Sung Ju Hwang. Representational continuity for unsupervised continual learning. In *International Conference on Learning Representations (ICLR)*, 2022.

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 (NeurIPS)*, 33:7308–7320, 2020.

Aristeidis Panos, Yuriko Kobe, Daniel Olmeda Reino, Rahaf Aljundi, and Richard E Turner. First session adaptation: A strong replay-free baseline for class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pp. 18820–18830, 2023.

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.

Quang Pham, Chenghao Liu, and Steven Hoi. Continual normalization: Rethinking batch normalization for online continual learning. *arXiv preprint arXiv:2203.16102*, 2022.

Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *In Proceedings of the European Conference on Computer Vision (ECCV)*, 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 (CVPR)*, 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.

Joshua T Vogelstein, Jayanta Dey, Hayden S Helm, Will LeVine, Ronak D Mehta, Tyler M Tomita, Haoyin Xu, Ali Geisa, Qingyang Wang, Gido M van de Ven, et al. Representation ensembling for synergistic lifelong learning with quasilinear complexity. *arXiv preprint arXiv:2004.12908*, 2020.

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

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 139–149, 2022.

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 (CVPR)*, pp. 374–382, 2019.

Lu Yu, Bartlomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6982–6991, 2020.

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 (ICML)*, pp. 12310–12320. PMLR, 2021.

Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. Slca: Slow learner with classifier alignment for continual learning on a pre-trained model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pp. 19148–19158, 2023.