# Pseudo-Labeling for Class Incremental Learning

Alexis Lechat[12]
alexis.lechat@onera.fr
Stéphane Herbin[1]
stephane.herbin@onera.fr
Frédéric Jurie[2]
frederic.jurie@unicaen.fr

[1] DTIS, ONERA
Université Paris-Saclay
FR-91123 Palaiseau, France

[2] Normandie Univ
UNICAEN, ENSICAEN, CNRS, GREYC
14000 Caen, France

**Abstract**

Class Incremental Learning (CIL) consists in training a model iteratively with limited amount of data from few classes that will never be seen again, resulting in catastrophic forgetting and lack of diversity. In this paper, we address these phenomena by assuming that, during incremental learning, additional unlabeled data are continually available, and propose a Pseudo-Labeling approach for class incremental learning (PLCiL) that makes use of a new adapted loss. We demonstrate that our method achieves better performance than supervised or other semi-supervised methods on standard class incremental benchmarks (CIFAR-100 and ImageNet-100) even when a self-supervised pre-training step using a large set of data is used as initialization. We also illustrate the advantages of our method in a more complex context with fewer labels. The code is available at https://github.com/alechat/PLCiL.

# 1 Introduction

Natural vision systems learn in a continuous way, benefiting from their constant interaction with the environment. Their skills are dynamically updated and accumulated throughout their life. While artificial models such as Deep Neural Networks (DNNs) have now achieved similar or even better performance in several perceptual tasks, their learning process is fundamentally different and relies mainly on supervised *batch training* which requires a large amount of annotated data.

Incrementally training artificial DNNs from an incoming data stream suffers from *catastrophic forgetting* [10, 12]: previously learned skills tend to be less accurate when new ones are integrated in the system. Continual Learning (CL) explores solutions to alleviate this phenomenon. It can be seen as finding a way to solve a *plasticity-stability dilemma* [27]: the model should be flexible enough to dynamically expand its knowledge (plasticity) while ensuring the integrity of previously accumulated knowledge (stability).

In this work, we focus on the *Class Incremental Learning* (CIL) scenario applied to image recognition: new classes gradually appear from the data stream and the total number of categories is not known beforehand. Many studies emulate a data stream by splitting a

© 2021. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

classification dataset into disjoint batches of several classes [7, 31]. Using this definition, we can see class incremental as analogous to a succession of small-scale supervised batch sessions, with each one learning a small subset of disjoint classes.

There is a large consensus that learning good visual representations is crucial to competitive classification performance [2]. Optimizing a feature extractor via *representation learning* or pre-training with an auxiliary task becomes mandatory when working with datasets of limited size. In particular, *self-supervised visual representation learning* [4, 8, 36] now achieves performance close to fully supervised methods while requiring less labeled data.

In a CL setting, however, methods struggle to learn good representations: this is especially true at the beginning of the learning process, since only a fraction of the data is available at each session. Training a DNN from scratch in a continuous framework requires the feature extractor to be constantly adjusted, resulting in unstable representations. This explains why some authors have taken the easier option of having a large number of classes at the beginning (e.g. 50 classes out of a total of 100) in order to consolidate the representations even before starting the incremental learning of the remaining classes [9, 17, 28].

In this paper, we aim to learn a model truly from scratch, using a semi-supervised representation learning mechanism, assuming that unlabeled data is available throughout the learning process. Note that the idea of using unlabeled data in this context has already been considered in [24, 43].

The underlying intuition behind this proposition is that, with an ideal representation space, the solution to class incremental reduces to allocating unassigned regions in the representation space to the new classes without needing to strongly modify the previous regions. While the advantages expressed in the representation learning literature directly transfer to CL, we also study how semi-supervision provides an answer to the plasticity-stability dilemma. We propose to exploit a process based on pseudo-labeling as a way to combine self-supervision with CIL.

Our contributions are threefold: i) We introduce a mechanism of *Pseudo-Labeling for Class incremental Learning* (PLCiL) and show its benefit in an original learning scheme adapted to semi-supervised CIL that combines 3 losses: a supervised loss, a self-supervised loss using pseudo-labels and a new distillation loss that ensures prediction consistency during CIL sessions. ii) Using PLCiL, we demonstrate that a self-supervision provides regularization against catastrophic forgetting, reaching state-of-the-art performance on class incremental benchmarks. iii) We propose a new class incremental evaluation protocol with even fewer labeled images. We show that our method can still learn the continual task thanks to the semi-supervision while fully supervised CL approaches can hardly compete on such data-scarce problems.

## 2 Related Work

Our proposed method borrows ideas from two different areas of the literature: that of continual learning and that of representation learning with self-supervision.

**Continual Learning [27]** refers to different settings [18, 37] such as CIL, which is the focus of this paper, or Task Incremental (TI) learning. The big challenge of CIL is to propose methods that are as insensitive to catastrophic forgetting during learning. The main approaches are described in the following.

*Rehearsal / Replay* consists in replaying part of the old data, stored in an episodic memory, and mixes it with new data during learning [6, 13, 19, 26, 31]. The best performing

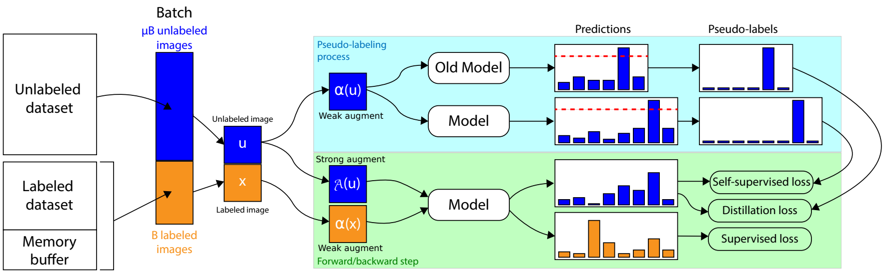


Figure 1: Overview of the proposed incremental training process. The upper part (blue box) represents the pseudo-labeling process dedicated to the automatic generation of a label for an unlabeled *u* image. The lower part (green box) is a standard supervised forward/backward process, including a 3-term loss function: i) standard supervised cross-entropy, ii) regularization between sessions by distillation, iii) regularization by self-supervision.

approaches follow this strategy, but their success depends on the number and representativeness of the chosen examples and require a large memory footprint. The limited amount of examples stored induces an imbalance between old classes and newly introduced ones. Models learning with rehearsal are then heavily biased. [14, 17, 39, 44] proposed solutions to compensate for the bias at the classifier level. Rather than simply storing old examples, an alternative is to produce data online by a continuously learned generative model [35, 38].

Another strategy is to make the classifier have outputs close to those of the previous sessions, especially on old data. To this end, Learning without Forgetting [25] has adapted the *Knowledge Distillation (KD)* loss introduced in [16]. The old classifier is the teacher and distills his knowledge to the new classifier, seen as the student. Most of the rehearsal based methods make use of some form of KD [5, 24, 31, 39, 44].

Other interesting strategies are *Parameter Control* [1, 7, 22, 42], *Dynamic Architectures* [29, 34, 40], *Generative Replay* [35, 38] or *Meta-Learning* [20, 30, 32].

Two approaches [24, 43] make the same hypothesis as our method: a large amount of unlabeled data available during the learning sessions. They both implement a self-supervised task: since KD does not require any ground-truth, they leverage unlabeled data by distilling knowledge from two teachers, a model expert on the old classes and a model trained only on the new classes, into a global model performing on all classes. While this kind of self-supervision is an efficient regularization against catastrophic forgetting, these methods do not exploit these additional data to enhance their representations.

Our method belongs to the rehearsal with memory category, but introduces both a new KD scheme and a self-supervised objective that aims to learn better representations throughout the CIL process.

**Self-Supervised Representation Learning** refers to a particular set of representation learning methods that use *pretext tasks*. Pretext tasks are used to automatically create artificial labels from unlabeled data that can be used to compute an error signal for learning.

A wide range of pretext tasks has been explored in the literature. We refer the reader to [21] for a complete review of the field. We do not detail this literature here because it is less central to our problem.

Our primary objective is to introduce regularization mechanisms against catastrophic forgetting, and we focus on semi-supervised methods [6, 46] that we believe are good candidates. Among the most powerful recent methods, [36] combines coherence regularization (robustness to transformations) and pseudo-labeling (semi-supervised learning).

Our approach proposes an original way to combine pseudo-labeling with CIL by introducing a specific KD loss ensuring prediction consistency between learning sessions.

# 3 Pseudo-Labeling for Class incremental learning

The PLCiL method proposed in this paper is based on a standard class incremental paradigm that relies on rehearsal learning with episodic memory [31]. In the classical setting, the available labeled samples during a training session only come from the memory and from the new annotated data. As in [24, 43], we propose to complement this paradigm with the possibility of using unlabeled data, with the motivation that access to unlabeled data at training time is easy (inexpensive) and does not violate the principle of exclusive annotation vocabulary between sessions that typifies class incremental learning.

We present this method in 4 steps: i) its overview (notations, training sessions, prediction model), ii) how the data is organized for each training session, iii) the various loss functions at the heart of our approach, and iv) a discussion and justification of its main components.

## 3.1 Class Incremental Learning

The class incremental scenario formalizes the incoming labeled data as a stream of subsets $\mathcal{X} = \{X^1, X^2, \dots, X^j, \dots\}$ where each $X^j = \{x_1^j, \dots, x_{n_j}^j\}$ only contains instances of class $j$. An incremental learning session uses several subsets pooled together and submitted to the network for training. Once the session is done, the pool of data is discarded and the associated classes will not appear again in the data stream. During the $i$-th session, the model is trained with the pool $T_i = \{X^{(i-1)s+1}, \dots, X^{is}\}$ with $s$ being the incremental step. For practical reasons and without loss of generality, we set $s$ fixed during the whole training process.

In our approach, we also have access to another source of data $\mathcal{U}$ providing *unlabeled* images belonging to the same domain as $\mathcal{X}$. We make the hypothesis that $\mathcal{U}$ is available to the process with no restriction at any time, although in practice only a limited quantity of data can be exploited during each learning session.

At each session $i$, the learning process uses a deep neural network with parameters $\Theta_i$ capable of predicting the class probability for any element $y \in \mathcal{Y}_i = \{y_1, \dots y_{i \times s}\}$ and any input sample $x \in \mathbb{R}^n$: $p(y = j|x) = f_j^i(x; \Theta_i)$. The DNN consists in a convolutional part with parameters $\theta_i$, which can be seen as a feature extractor $\phi(x, \theta_i) : \mathbb{R}^n \to \mathbb{R}^d$, followed by a fully connected layer classifier with parameters $\mathbf{w}_i \in \mathbb{R}^{d \times (i \times s)}$. At each session, the model is initialized with the previous set of parameters $\Theta_{i-1} = (\theta_{i-1}, \mathbf{w}_{i-1})$. $s$ outputs are added to the single-head classifier while the encoder keeps the same parameter structure in $\theta_i$.

## 3.2 Buffer management and training data

The training data at each session comes from four different sources: the pool of annotated data $T_i$, the unlabeled data $\mathcal{U}$, the rehearsal buffer $\mathcal{B}$ and a data augmentation process.

**Memory Buffer** $\mathcal{B}$ Its role is to store old annotated samples to mimic an episodic memory. It is characterized by a hyperparameter $K$ that defines the number of stored samples. During the learning process, we use random selection to pick the exemplars while ensuring the balance between classes, i.e. after the $i$-th session, $\mathcal{B}$ contains $\lfloor \frac{K}{i \times s} \rfloor$ exemplars per class.

**Data Augmentation** We will see in the next section that our approach makes use of the generation of pseudo-labels for non-annotated images. This mechanism is based on the idea that images must keep the same pseudo-labels even when a transformation is applied to them.

We define two types of possible transformations: strong and weak, denoted respectively as $\mathcal{A}(.)$ and $\alpha(.)$. In practice, our weak transformations consist of random horizontal flips and translations, as it is practiced in most representation learning methods [31, 39]. Our strong transformations strictly follow the implementation of [36]: they include transformations such as cutout, translation, rotation, color and brightness adjustment, etc. and use the CTAugment sampling strategy described in [3]. The complete list of augmentations is provided in appendix D along with an ablation evaluating the combination of both weak and strong augmentations for consistency regularization.

## 3.3 Learning process

The parameters $\Theta_i$ are learned by stochastic gradient descent (SGD) at each session $i$. As is standard when applying SGD, each parameter update step makes use of a mini-batch of data which is randomly sampled at each step. In the semi-supervised scheme proposed in our approach, such a mini-batch $S$ is made of two types of data: a subset $S_l$ containing $B$ labeled images sampled from $T_i \cup \mathcal{B}$ and another subset $S_u$ composed of $\mu B$ images from $\mathcal{U}$ where $\mu$ is a scalar hyperparameter.

The proposed PLCiL algorithm relies on the optimization of 3 combined losses targeting 3 different objectives: i) supervision coming from the novel labeled data, ii) consistency regularization using pseudo-labels automatically generated on the unlabeled data, iii) self-supervised knowledge distillation adding extra regularization. We describe these 3 losses in the following, removing the session number $i$ in the notations when its reference is not necessary. The overall training process is illustrated in Figure 1.

**Supervised loss $l_{\text{sup}}$.** Its role is to use labeled data to learn the model. To improve robustness, a data augmentation step is introduced using a weak transformation $\alpha$. The supervised signal is back propagated to the network using the standard cross-entropy loss between the output of the DNN and the true labels. It can be expressed as:

$$l_{\text{sup}} = \frac{1}{B} \sum_{(x,y)\in S_l} H(y, f(\alpha(x);\Theta)) \tag{1}$$

where $f(x;\Theta) \in \mathbb{R}^{|\mathcal{Y}|}$ is the predicted class distribution given an input $x$, $|\mathcal{Y}|$ is the number of classes considered in the current session. $H$ is the cross-entropy defined by:

$$H(y, f(\alpha(x);\Theta)) = -\sum_{j\in\mathcal{Y}} y_j \log(f_j(\alpha(x);\Theta)) \tag{2}$$

where $y$ is a one-hot encoding of the true class label.

**Self-Supervised loss $l_{\text{self}}$.** Its role is to regularize the image representations by mimicking true annotation using pseudo-labels on unlabeled data $u$ from $S_u$ and two levels of image transformations: weak and strong. A weakly transformed data $\alpha(u)$ is first fed to the DNN. If the model is confident enough on its output (according to a threshold $\tau$ on the scores), this prediction is used as a pseudo-label for a cross-entropy loss on the strongly augmented image $\mathcal{A}(u)$. Given the prediction on weakly augmented data $q_u = f(\alpha(u), \Theta)$ and the pseudo-label

$\hat{q}_u = \text{argmax}(q_u)$, the resulting self-supervised loss is:

$$l_{\text{self}} = \frac{1}{\mu B} \sum_{u \in S_u} \mathbb{1}_{\max(q_u) > \tau} H(\hat{q}_u, f(\mathcal{A}(u); \Theta_i)) \tag{3}$$

**Distillation loss $l_{\text{kd}}$.** Distillation is used to ensure prediction consistency between sessions and is thus expected to lower forgetting. We again use pseudo-labeling with confidence thresholding but with the difference that the pseudo-labels are generated using the model from the previous session $f(x; \Theta_{i-1})$. Let $q_{\text{old}} = f(\alpha(u), \Theta_{i-1})$ and $\hat{q}_{\text{old}} = \text{argmax}(q_{\text{old}})$ be respectively the prediction and the associated pseudo-label at the previous session. The knowledge distillation loss is defined as:

$$l_{\text{kd}} = \frac{1}{\mu B} \sum_{u \in S_u} \mathbb{1}_{\max(q_{\text{old}}) > \tau} H(\hat{q}_{\text{old}}, f(\mathcal{A}(u); \Theta_i)) \tag{4}$$

Note that in our approach, $l_{\text{kd}}$ is computed only on unlabeled samples.

**Total loss.** PLCiL combines the 3 training objectives during the optimization process:

$$loss = l_{\text{sup}} + \lambda (l_{\text{self}} + \eta \, l_{\text{kd}}) \tag{5}$$

with $\eta = \frac{|\mathcal{Y}_{i-1}|}{|\mathcal{Y}_i|}$ the ratio between the number of classes learned by the old model and the current number of classes. This scalar is used in [25, 39, 44] to balance the distillation loss. Note that with the assumption of $s$ constant, $\eta = \frac{i-1}{i}$. $\lambda$ is a scalar hyper-parameter balancing supervision and self-supervision.

## 3.4 Discussion

The design of our algorithm was guided by 3 key objectives: i) the use of additional unlabeled data to improve and make the learned representations more stable, due to the visual diversity they provide; ii) the use of unlabeled data to add self-regularization to the classification head; iii) the use of pseudo-labels, generated by the model of the previous session to distill knowledge between incremental steps, adding additional regularization.

One of our main contributions is therefore knowledge distillation via unlabeled data. This is related to consistency regularization by matching the distribution of outputs of the two models (the old and the new one), while in related work KD is usually based on soft sharpening [25]. Pseudo-labeling KD is very specific to our approach as we use distillation on unlabeled samples: a confidence threshold allows to select the nature of the distilled knowledge, retaining only the relevant examples, while classical distillation [24, 39, 43, 44] blindly transfers a fraction (temperature parameter) of the whole knowledge contained in all examples. This, together with the fact that the classification head grows with each training session, makes our problem very different from that of FixMatch [36].

The argmax-based distillation loss $l_{\text{kd}}$, defined in Eq. (4), takes advantage of both the unlabeled mini-batch $S_u$ and the old stored model $f(x; \Theta_{i-1})$. In addition to the self-supervision loss ($l_{\text{self}}$) which regularizes the model as such, $l_{\text{kd}}$ enforces the consistency between sessions (i.e., between $\Theta_{i-1}$ and $\Theta_i$).

During the learning process, the model of the previous session is expert for the classes already seen. During a new session it can produce pseudo-labels for unlabeled images. In practice, even if these images come from categories never seen, they have links with the categories seen because of the selection process (application of transformation and score thresholding), thus creating bridges between sessions. This is very different from pseudo-labeling for semi-supervised learning, which concerns already known classes.

This distillation mechanism also makes it possible to properly take into account the increase in the number of outputs of the classification head. The sudden increase in the number of logits overwhelms the output distribution, due to the softmax activation. Thus, during the first training epochs of $f(x;\Theta_i)$, almost none of the predictions on unlabeled images will exceed the $\tau$ threshold (usually set close to 1), which means that the pseudo-labeling process is reset at each session. The KD loss ensures that consistent pseudo-labeling is maintained over the sessions. The $\eta$ parameter takes into account the fact that the pseudo-labels can change over time as the number of possible outputs increases. When the teacher (i.e. the previous model) knows only a few classes, many images are likely to get a wrong pseudo-label, so less weight is given to the labels provided by the previous model in favor of the pseudo-labeling done by the current model. However, a ratio $\eta$ close to 1 means that the two models know about the same amount of classes. We can assume that, in this case, most of the images can be correctly labeled by both and then give the same credit to $l_{\text{kd}}$ and $l_{\text{self}}$.

# 4 Experiments

We experimentally validated our method on the 2 datasets commonly used to evaluate class incremental methods, namely CIFAR-100 [23] and ImageNet-100 [33]. ImageNet-100 is a subset of ImageNet-1000 where only 100 classes are considered (same classes as [5, 31]).

Evaluation is done using the standard incremental accuracy metric: the model is tested at the end of each session on all the classes seen so far. The Last Accuracy is measured on all classes once the class incremental process is completed, and the Average Accuracy is the mean of all incremental accuracies, excluding the first session which cannot be considered as incremental. For ImageNet experiments, we reported the top-5 accuracy. We averaged 3 runs for CIFAR-100 and 1 for ImageNet, as it is commonly done in the literature [39, 44].

Note that exploiting additional unlabeled data – which is the main objective of the proposed approach – is not directly feasible with existing methods since they are designed to work under full supervision. Only DMC+ [43] is designed to work under the same scenario. Thus, the results, more than a direct raw comparison between the methods, should be seen as a showcase of the advantages of using semi-supervision when training a CIL model.

We nevertheless compare our method to the following (fully supervised) rehearsal-based solutions which are known to works in large scale CIL scenarios: GDumb [28], iCaRL [31], BiC [39], Weight Aligning (WA) [44] and, as mentioned above, with the semi-supervised DMC+ method [43]. To give fully supervised methods a fairer chance, we propose some experiments in which they are pre-trained with the same unlabeled data. We have re-implemented all the competing methods so as to have exactly the same backbone network and the same DL framework for all the methods being compared. Due to encountered difficulties in reproducing Global Distillation (GD) [24] and its costly requirements for unlabeled data sampling, we directly report results from the original paper when possible.

All methods are tested using the same DNN : a Wide-ResNet-28-8 [41] (WRN28-8) for CIFAR and a ResNet-18 [15] for ImageNet. For reference, WRN28-8 achieves 82.8% on CIFAR-100 and ResNet-18 top-5 accuracy on ImageNet-100 is 94.4%. All hyperparameters used are detailed in appendix A.1 with an in depth study of the sensitivity of $\mu$, $\tau$ and $\lambda$ in appendix C. For CIFAR-100, WRN28-8 contains many more parameters than ResNet-32 commonly used by CIL methods. This choice of backbone is discussed in appendix A.2.

| Method | Last (%) | Avg (%) |
|---|---|---|
| GDumb [28] | 27.8 | 42.0 |
| iCaRL [31] | 53.9 | 63.9 |
| BiC [39] | 55.9 | 67.1 |
| WA [44] | 50.8 | 64.4 |
| DMC+ [43] | 50.4 | 62.8 |
| GD [24] | 54.9* | 68.1* |
| Ours | **61.5** | **74.0** |

* Results from [24] with WRN16-2 backbone and Tiny Images (80M images) as unlabeled data.

Table 1: CIL comparison on CIFAR-100-full.

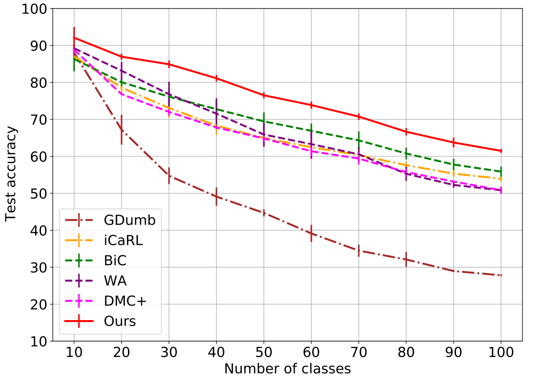


Figure 2: Plot of incremental accuracy over the successive sessions on CIFAR-100-full. Best viewed in PDF.

| Method | CIFAR-100-20% | | | | ImageNet-100-10% | | | |
|---|---|---|---|---|---|---|---|---|
| | Last (%) | Avg (%) | Last (%) | Avg (%) | Last (%) | Avg (%) | Last (%) | Avg (%) |
| | Random Init | | RotNet Init | | Random Init | | RotNet Init | |
| GDumb [28] | 28.2 | 42.2 | 25.3 | 40.4 | 40.6 | 59.6 | 43.7 | 62.0 |
| iCaRL [31] | 42.7 | 48.9 | 43.9 | 51.1 | 45.4 | 57.8 | 51.9 | 59.5 |
| BiC [39] | 43.3 | 49.8 | 43.5 | 57.3 | 50.7 | 62.4 | 52.2 | 68.8 |
| WA [44] | 40.5 | 49.7 | 45.5 | 55.4 | 30.2 | 54.7 | 40.9 | 64.5 |
| DMC+ [43] | 36.4 | 42.8 | 39.3 | 49.8 | 56.2 | 68.1 | 57.5 | 69.6 |
| Ours | **59.8** | **66.5** | **59.5** | **67.6** | **61.3** | **73.8** | **61.2** | **75.0** |

Table 2: Experiments with fewer labels available: CIFAR-100-20% and ImageNet-100-10%.

## 4.1 Class Incremental Results

We conducted experiments following the standard class incremental protocol [5, 31] on CIFAR-100. We set $s = 10$, i.e. 10 sessions of 10 classes. The memory size for rehearsal $K$ is set to 2000. Since we use the whole labeled dataset here, we refer to this experimental setting as CIFAR-100-full. For DMC+ and PLCiL, we emulate an unlabeled data stream by randomly sampling $100K$ unlabeled images from ImageNet-1000 (subsampled to $32 \times 32$).

The results obtained are available in Table 1. They show the value of exploiting unlabeled data in a joint CIL framework. Indeed, our PLCiL method consistently outperforms other state-of-the-art methods on CIFAR-100-full, gaining +5.6% on final accuracy. In the following section, we demonstrate that the semi-supervised nature of our method is able to address even more challenging scenarios where less supervision is available.

## 4.2 Semi-Supervised Class Incremental Results

The following experiments study the behavior of our method in scenarios even closer to the non semi-supervised scenarios: only a very limited set of labeled data is available, while a large amount of cheap unlabeled data is accessible.

Following this principle, we keep the previous settings with $s = 10$ but reduce the size

| Scenario | Last (%) | Avg (%) |
|---|---|---|
| a. ImageNet-900 | 61.3 | 73.8 |
| b. ImageNet-100 | 76.9 | 83.3 |
| c. ImageNet-1000 | 65.1 | 76.2 |
| d. Places365 | 59.6 | 72.7 |

Table 3: Class incremental performance on ImageNet-100-10% with 4 different unlabeled datasets.

| Loss | Last (%) | Avg (%) |
|---|---|---|
| $l_{sup}$ | 44.1 | 59.8 |
| $l_{sup} + \lambda \eta l_{kd}$ | 61.9 | 63.9 |
| $l_{sup} + \lambda l_{self}$ | 50.3 | 65.15 |
| $l_{sup} + \lambda (l_{self} + \eta l_{kd})$ | 61.5 | 74.0 |
| $l_{sup} + \lambda (l_{self} + \eta l_{standkd})$ | 52.0 | 65.6 |
| $l_{sup} + \lambda (l_{soft} + \eta l_{standkd})$ | 50.9 | 63.4 |

Table 4: CIL on CIFAR-100-full with only specific components of the loss enabled.

of the labeled dataset. For CIFAR-100-20%, we randomly pick 100 samples per class out of the 500, while for ImageNet-100-10%, 130 labeled samples per class are retained. This is in line with the amount of labels commonly used in semi-supervised works [36]. The memory budget remains unchanged with $K = 2000$.

The process of collecting the unlabeled data is the same as before: we sample $100,000$ data from ImageNet-1000 at the beginning of each session. To avoid any data leakage, images belonging to classes learned incrementally are removed from the unlabeled data.

Since this setting is very difficult for fully supervised methods due to the scarcity of data, we also compare with a self-supervised initialization using the same amount of unlabeled data (1M images). WRN28-8 and ResNet-18 were trained using RotNet [11] on ImageNet-1000 (with classes to be learned incrementally excluded). Self-supervised pre-training is the most immediate way to allow fully supervised methods to access as much unlabeled data as PLCiL or DMC+.

The results are presented in Table 2. The lack of data is clearly perceptible for supervised approaches and only GDumb is stable due to the fact that its performance only depends on the buffer size. Meanwhile, PLCiL maintains the level of performance obtained during the first protocol, proving that in scenarios with very few labels, our method can efficiently exploit unlabeled data. Although the representations obtained by self-supervision are better, the exploitation of unlabeled data throughout is more efficient with PLCiL, which is outperforming all pretrained fully supervised methods.

PLCiL is consistently better than DMC+, which however exploits the same quantity of data as PLCiL. We believe this is because DMC+ only exploits unlabeled data to distill knowledge from a learned model with full supervision. From the plasticity-stability dilemma, such approach focuses on leveraging unlabeled data to improve stability. The performance on new classes is dependent on the fully supervised training phase of the teacher model. While PLCiL has a similar behavior with $l_{kd}$, it also aims to improve plasticity by also implementing self-supervised representation learning with $l_{self}$ and its data augmentation strategy. The ablation study gives more details on the contribution of each loss.

## 4.3 Ablation study

**Impact of the unlabeled data source.** In the experiments reported in tables 1 and 2, we use an unlabeled dataset which is semantically similar to the target while excluding all samples belonging to the learned classes (as it is common in self-supervised literature). This emulates an application where both the unlabeled data $\mathcal{U}$ and labeled data $\mathcal{X}$ are collected from the same environment.

In order to evaluate the influence of the unlabeled source, we repeat the same experiment on ImageNet-100-10% with random initialization using 4 different unlabeled datasets for $\mathcal{U}$: (a) samples from the remaining 900 classes as in section 4.2. (b) samples from the remaining images of ImageNet-100 (which is feasible since only 10% of the labeled images are used in $\mathcal{X}$). (c) samples from all unused images from ImageNet-1000 so that $\mathcal{U}$ contains about one tenth of images belonging to classes in $\mathcal{X}$ using uniform sampling. (d) samples from a semantically unrelated dataset (Places-365 [45]).

Results are presented in Table 3. (c) shows that a reasonable class leakage between $\mathcal{U}$ and $\mathcal{X}$ is efficiently leveraged by our model, improving the performance from our baseline (a). We also tried the idealistic scenario of (b) with the exact same classes in both dataset. This provides an upper bound of our method given a fully related unlabeled dataset. At last, using a semantically unrelated dataset in (d) slightly lowers the performance compared to (a) but still outperforms the other approaches shown in table 2.

**Contribution of each loss component** We ran several variants of PLCiL on CIFAR-100-full to evaluate the contribution of each loss term (see Table 4). We noticed two complementary behaviors when $l_{kd}$ and $l_{self}$ are used separately. The version with pseudo-labeling KD focuses on the stability of the model, keeping the most consistent accuracy from session to session and achieves the best final accuracy despite a lower average. This is due to the fact that it struggles to learn new classes, especially during the early stages where the proportion of old classes is low, making KD less relevant. The version with only $l_{self}$ enhances the plasticity of the model, allowing to learn the new classes with better accuracy as it is shown in the first sessions. However, this variant still lacks regularization to alleviate the catastrophic forgetting during the later stages. By combining both, PLCiL optimizes the *plasticity-stability* trade-off and gives satisfactory results during all sessions. The full version of Table 4 with accuracy at all sessions is provided in Appendix B.1.

**Efficiency of Pseudo-Labeling for Knowledge Distillation.** We replaced our $l_{kd}$ by a standard KD, as in [39, 44], but still applied on unlabeled data only. The results are given in the penultimate line of Table 4. Standard KD ($l_{standkd}$) has little or no effect and gives results similar to BiC and WA. This comparison highlights the efficiency of our custom distillation with only unlabeled data and pseudo-labeling.

We also experimented with the removing of the Pseudo-Labeling step. As for KD, the *soft* alternative consists in using directly the output distribution as target. Using a similar loss to the standard distillation $l_{soft}$, we apply consistency regularization between weakly and strongly augmented images. In the last line of Table 4 with both $l_{soft}$ and $l_{standkd}$, i.e. with the thresholding removed in all losses, the performance further drop. This result validates the usefulness of a hard thresholding for leveraging unlabeled data.

## 5 Conclusion

We introduce a simple class incremental method combining learning by rehearsal and self-supervised learning that takes advantage of complementary unlabeled data during the learning process. With this simple approach, we demonstrate that semi-supervision is a valuable aid to the problems of catastrophic forgetting and scarcity of data at training time. Experimental validation on both CIFAR-100 and ImageNet-100 shows that PLCiL has better learning capacities than existing methods, thanks to finer and more stable representations.

# References


[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory Aware Synapses: Learning what (not) to forget. In *European Conference on Computer Vision, ECCV*, 2018.

[2] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, 2013.

[3] David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. ReMixMatch: Semi-Supervised Learning with Distribution Matching and Augmentation Anchoring. In *8th International Conference on Learning Representations, ICLR*, 2020.

[4] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2020.

[5] Francisco M. Castro, Manuel J. Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-End Incremental Learning. In *European Conference on Computer Vision, ECCV*, 2018.

[6] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. *Semi-Supervised Learning*. The MIT Press, 2006. ISBN 9780262033589.

[7] Arslan Chaudhry, Puneet Kumar Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence. In *European Conference on Computer Vision, ECCV*, 2018.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *37th International Conference on Machine Learning, ICML*, 2020.

[9] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. PODNet: Pooled Outputs Distillation for Small-Tasks Incremental Learning. In *European Conference on Computer Vision, ECCV*, 2020.

[10] Robert M. French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128 – 135, 1999. ISSN 1364-6613.

[11] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised Representation Learning by Predicting Image Rotations. In *6th International Conference on Learning Representations, ICLR*, 2018.

[12] Ian J. Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An Empirical Investigation of Catastrophic Forgetting in Gradient-Based Neural Networks. *arXiv:1312.6211*, 2013.

[13] Tyler L. Hayes, Kushal Kafle, Robik Shrestha, Manoj Acharya, and Christopher Kanan. REMIND Your Neural Network to Prevent Catastrophic Forgetting. In *European Conference on Computer Vision, ECCV*, 2020.

[14] Jiangpeng He, Runyu Mao, Zeman Shao, and Fengqing Zhu. Incremental Learning in Online Scenario. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 2020.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016.

[16] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the Knowledge in a Neural Network. *arXiv:1503.02531*, 2015.

[17] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a Unified Classifier Incrementally via Rebalancing. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2019.

[18] Yen-Chang Hsu, Yen-Cheng Liu, and Zsolt Kira. Re-evaluating Continual Learning Scenarios: A Categorization and Case for Strong Baselines. *arXiv:1810.12488*, 2018.

[19] Ahmet Iscen, Jeffrey Zhang, Svetlana Lazebnik, and Cordelia Schmid. Memory-Efficient Incremental Learning Through Feature Adaptation. In *European Conference on Computer Vision, ECCV*, 2020.

[20] Khurram Javed and Martha White. Meta-Learning Representations for Continual Learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2019.

[21] Longlong Jing and Yingli Tian. Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey. *arXiv:1902.06162*, 2019.

[22] 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.

[23] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, 2009.

[24] Kibok Lee, Kimin Lee, Jinwoo Shin, and Honglak Lee. Overcoming Catastrophic Forgetting With Unlabeled Data in the Wild. In *International Conference on Computer Vision, ICCV*, 2019.

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

[26] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics Training: Multi-Class Incremental Learning Without Forgetting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 2020.

[27] 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.

[28] Ameya Prabhu, Philip Torr, and Puneet Dokania. GDumb: A Simple Approach that Questions Our Progress in Continual Learning. In *The European Conference on Computer Vision (ECCV)*, 2020.

[29] Jathushan Rajasegaran, Munawar Hayat, Salman H. Khan, Fahad Shahbaz Khan, and Ling Shao. Random Path Selection for Continual Learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2019.

[30] Jathushan Rajasegaran, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Mubarak Shah. iTAML: An Incremental Task-Agnostic Meta-learning Approach. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 2020.

[31] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. iCaRL: Incremental Classifier and Representation Learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[32] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to Learn without Forgetting by Maximizing Transfer and Minimizing Interference. In *7th International Conference on Learning Representations, ICLR*, 2019.

[33] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[34] 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 *35th International Conference on Machine Learning, ICML*, 2018.

[35] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual Learning with Deep Generative Replay. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2017.

[36] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS*, 2020.

[37] Gido M. van de Ven and Andreas S. Tolias. Three scenarios for continual learning. *arXiv:1904.07734*, 2019.

[38] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, Zhengyou Zhang, and Yun Fu. Incremental Classifier Learning with Generative Adversarial Networks. *arXiv:1802.00853*, 2018.

[39] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large Scale Incremental Learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2019.

[40] Shipeng Yan, Jiangwei Xie, and Xuming He. DER: Dynamically Expandable Representation for Class Incremental Learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2021.

[41] Sergey Zagoruyko and Nikos Komodakis. Wide Residual Networks. In *Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016*, 2016.

[42] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual Learning Through Synaptic Intelligence. In *34th International Conference on Machine Learning, ICML*, 2017.

[43] Junting Zhang, Jie Zhang, Shalini Ghosh, Dawei Li, Serafettin Tasci, Larry P. Heck, Heming Zhang, and C.-C. Jay Kuo. Class-incremental Learning via Deep Model Consolidation. In *IEEE Winter Conference on Applications of Computer Vision, WACV*, 2020.

[44] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining Discrimination and Fairness in Class Incremental Learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 2020.

[45] Bolei Zhou, Àgata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 Million Image Database for Scene Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(6):1452–1464, 2018.

[46] Xiaojin Zhu and Andrew B. Goldberg. *Introduction to Semi-Supervised Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2009.