# Prototype-Sample Relation Distillation: Towards Replay-Free Continual Learning

Nader Asadi [1 2]   MohammadReza Davari [1 2]   Sudhir Mudur [1]   Rahaf Aljundi [3]   Eugene Belilovsky [1 2]

## Abstract

In Continual learning (CL) balancing effective adaptation while combating catastrophic forgetting is a central challenge. Many of the recent best-performing methods utilize various forms of prior task data, *e.g.* a replay buffer, to tackle the catastrophic forgetting problem. Having access to previous task data can be restrictive in many real-world scenarios, for example when task data is sensitive or proprietary. To overcome the necessity of using previous tasks' data, in this work, we start with strong representation learning methods that have been shown to be less prone to forgetting. We propose a holistic approach to jointly learn the representation and class prototypes while maintaining the relevance of old class prototypes and their embedded similarities. Specifically, samples are mapped to an embedding space where the representations are learned using a supervised contrastive loss. Class prototypes are evolved continually in the same latent space, enabling learning and prediction at any point. To continually adapt the prototypes without keeping any prior task data, we propose a novel distillation loss that constrains class prototypes to maintain relative similarities as compared to new task data. This method yields state-of-the-art performance in the task-incremental setting, outperforming methods relying on large amounts of data, and provides strong performance in the class-incremental setting without using any stored data points.

## 1. Introduction

Continual Learning (CL) aims to continuously acquire knowledge from an ever-changing stream of data. The goal
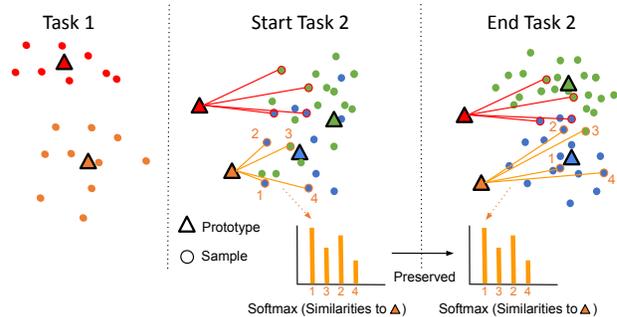


*Figure 1.* Illustration of our Prototype-Sample Relation Distillation (PRD). For each prior task prototype, we preserve the relative ordering of samples in the mini-batch. This allows the representation to adapt to new tasks while maintaining relevant positions of past prototypes. Illustrated for the prototype in orange 4 samples in the minibatch are ranked 1 through 4 based on similarity. PRD attempts to preserve this ranking while learning the new task.

of the learner is to continuously incorporate new information from the data stream while retaining previously acquired knowledge. Performance decay of the system on the older samples due to the loss of previously acquired knowledge is referred to as catastrophic forgetting (McCloskey & Cohen, 1989), which represents a great challenge in CL. Thus CL algorithms are typically designed to control for catastrophic forgetting while observing additional restrictions such as memory and computation constraints.

Some of the early work in modern continual learning such as LwF (Li & Hoiem, 2017) and EWC (Kirkpatrick et al., 2016) propose solutions that do not require storage of past data. However, in complex settings with long task sequences, these techniques tend to under-perform by a wide margin compared to the idealized joint training (Chaudhry et al., 2019). Many recent high-performing approaches in this area maintain existing samples in some form of buffer, allowing them to be reused for distillation(Rebuffi et al., 2017), replay (Chaudhry et al., 2019; Caccia et al., 2020), or as part of gradient alignment constraints (Lopez-Paz et al., 2017; Chaudhry et al.). These approaches have been shown to be more efficient and have become a predominant approach for many state-of-the-art continual learning systems (Verwimp et al., 2022). On the other hand in many cases when training on a new task it may be prohibited to store prior data. For

example, prior task data may be sensitive (e.g. medical data) or it may consist of proprietary data that is not aimed for release. Moreover, methods relying on prior task data tend to grow the storage with the number of tasks (Chaudhry et al.; Caccia et al., 2020), which can be prohibitive under severe storage constraints. Thus developing methods that can match or exceed the efficiency of data-storage-based methods is of great importance.

Recently (Davari et al., 2022) observed that for many continual learning tasks, the representational power of deep networks trained with naive fine-tuning can remain remarkably efficient for representing both new and old task data. In particular, it was observed that when performing continual learning with the Supervised Contrastive Loss (Khosla et al., 2020) and no CL constraints, the efficiency of representations on old task data tends to match that of complex CL data-storage methods. These observations relied on an oracle measure of the deep representations and did not provide a practical solution. In order to link the powerful representation learning to the effective prediction of prior class data we can consider alternatives for making the final prediction. An approach previously taken in the continual learning literature is to use the notion of class prototypes (De Lange & Tuytelaars, 2021), vector representations whose similarity to new sample representation can give predictions of the target class. If we take the observation that representations of old classes are already well separated (Davari et al., 2022) then an efficient continual learner can be obtained by simply maintaining correct estimates of past class prototypes.

In this work, we propose an effective mechanism to not only maintain relevant class prototypes but also leverage the knowledge embedded in these prototypes to further reduce representation forgetting. We combine contrastive representation learning with a prototype-based classifier. The new class prototypes are learned such that no direct negative influence is incurred on previous prototypes. Then, a novel loss formulation based on the relative similarity of new task data to old class prototypes is deployed to maintain the relevancy of old class prototypes while encouraging the learned representation to remain effective for old tasks. Our approach is illustrated in Fig. 1. Our proposed method, Prototype-Sample Relation Distillation (PRD), maintains the relative relation of each prototype, by minimizing changes in the softmax distribution over samples. This effectively allows representations to adapt to new classes while keeping prototypes from old classes relevant. We now summarize our *overall contributions* in this work:

- We propose a *novel* CL method, PRD, that *does not rely* on prior data storage during training or inference.

- In a variety of challenging settings (task and class-incremental), datasets (SplitMiniImagenet (Vinyals et al., 2016), SplitCIFAR100 (Krizhevsky et al., 2009), Imagenet-32 (Davari et al., 2022)), and task sequence lengths (20 to 200), we demonstrate that PRD leads to large improvements over both replay-based and replay-free methods.

- Throughout several experiments, we demonstrate that our method not only achieves strong control of forgetting of previously observed tasks but also leads to improved plasticity in learning new tasks.

In the following section, Sec. 2, we summarize the related work and then describe the essence of our proposed solution, Sec. 3. We demonstrate the effectiveness of our approach in Sec. 4, and conclude our work in Sec. 5. [1]

## 2. Related Work

The primary goal of many CL methods is to mitigate the catastrophic forgetting phenomenon while optimizing the forward and backward knowledge transfer between tasks is seen as a secondary objective. One branch of algorithms addresses the issue of catastrophic forgetting by modifying and growing the model architecture as new tasks are observed (Rusu et al., 2016; Aljundi et al., 2016; Li et al., 2019; Rosenfeld & Tsotsos, 2018). Under the fixed architecture constraints, the algorithms can be divided into two categories. The first and more popular branch is the rehearsal methods. These methods store and re-use samples of the past tasks while observing the new ones (Lopez-Paz et al., 2017; Chaudhry et al., 2019). The second family of approaches is the regularization-based methods. These methods preserve the previously learned information by imposing penalty terms on the objective of the new tasks, including popular methods such as LwF (Li & Hoiem, 2017) and EWC (Kirkpatrick et al., 2016; Chaudhry et al., 2018), where the former imposes a knowledge distillation penalty (Hinton et al., 2015) on the objective of the newly observed task and the latter a quadratic penalty based on Fisher information matrix (Myung, 2003).

Recently, several works have considered the use of SupCon loss (Khosla et al., 2020) in continual learning (Caccia et al., 2022; Asadi et al., 2022; Cha et al., 2021). These studies have been largely focused on the combination of SupCon loss (Khosla et al., 2020) with replay buffers in the online setting, and do not consider the notion of class prototypes in the replay-free setting. (Davari et al., 2022) demonstrated that the use of the SupCon loss in the offline setting yields more effective "representation forgetting" (forgetting as measured by an oracle training of a linear probe). However, a direct application of this observation of the SupCon loss was not proposed in this prior work.

---

[1]The code for our experiments is available at https://github.com/naderAsadi/CLHive.

(De Lange & Tuytelaars, 2020) proposed a prototype-based evolution strategy that continually updated prototypes using a momentum update combined with taking the mean of stored exemplars. Contrary to the present work this method focused on the online setting and leveraging stored data. Furthermore, it did not exploit the efficient and stable representation properties of contrastive supervised learning.

Knowledge distillation (Hinton et al., 2015), where a student network attempts to mimic the behavior of a teacher network is a popular technique in deep representation learning(Hinton et al., 2015; Tian et al., 2019; Zhu et al., 2021b), often used to reduce model size. Classical distillation techniques have been applied in various contexts in CL. (Rebuffi et al., 2017; Javed & Shafait, 2018) utilized a distillation loss alongside replayed examples, constraining the current model to give similar outputs. (Barletti et al., 2022) recently proposed to use a triplet loss alongside a contrastive distillation term. Here samples are constrained to have similar distances under current and previous models. By contrast, we apply a relation distillation that constrains the relative distances of old class prototypes to samples to be preserved.

Another application of classical distillation techniques closely related to our work (Wu et al., 2021) proposed a method that does not require storage of prior task data. This approach relies on constraining the distance between embeddings of the old and new models combined with a cross-entropy term. The constraint here can be analogous to a traditional distillation term, while our approach focuses on relation distillation to prototypes. (Wu et al., 2021) also utilizes a self-supervised learning objective based on the rotation of images to enhance the representation learning similar to (Zhu et al., 2021a).

Relation distillation has recently been used in teacher-student methods (Park et al., 2019). Unlike conventional knowledge distillation techniques that attempt to make student network representations similar to teacher networks, relation distillation maintains relative distances between a set of points. In the present work, we apply a related idea in the context of continual learning, maintaining relative relationships between prototypes and current task samples.

## 3. Methodology

We consider a general continual learning setting where a learner is faced with a possibly never-ending stream of data divided into separate training sessions. At each session $S_t$, a set of data $\mathbf{X}^t$ and their respective labels $\mathbf{Y}^t$ are drawn from a distribution $D_t$ characterized by $P(\mathbf{X}, \mathbf{Y}|T = t)$. When learning new sessions, it is assumed that access to the samples from previous sessions is restricted. This definition covers both task-incremental settings where $(\mathbf{X}^t, \mathbf{Y}^t)$ represent a separate task, the class-incremental scenario

where changes in $P(\mathbf{X})$ induces a shift on $P(\mathbf{Y})$, and the domain-incremental learning where changes in $P(\mathbf{X})$ does not affect $P(\mathbf{Y})$. We consider a neural network composed of an encoder $f$ that maps an input sample $x$ to its features representation $f(x) \in \mathbb{R}^d$ and a projection head $g$ that projects the features onto another latent space $g \circ f(x) \in \mathbb{R}^k$ where $k < d$. Our goal is to minimize the objective loss $\mathcal{L}_t$ on the new session data while not increasing the objective loss of the previously learned sessions $\mathcal{L}_i \,_{\forall i < t}$.

A common approach to control the loss of the previously encountered sessions is to use a buffer of stored samples and reuse them upon encountering new sessions. In our approach we do not require access to past data, instead, we approximate the behavior of the previously seen classes of objects via a set of prototypes. In our approach upon visiting a new session, we employ a novel distillation term to approximate the now-inaccessible loss of the previous sessions and set to restrict this surrogate loss in order to control the loss of the previously seen sessions. In the following sections, we introduce the different parts of the objective function used to optimize the model at each step.

**Supervised Contrastive Learning** Supervised Contrastive Learning (Khosla et al., 2020) is a powerful representation learning method observed to be useful in many downstream tasks. (Davari et al., 2022) employed Supervised Contrastive training for continual learning and showed that the learned representations are less prone to forgetting compared to that learned with Cross Entropy loss (CE). In this work, we build on this observation and propose a solution to jointly train the representation and classification head in an incremental fashion. In order to optimize the representation for the task being learned, we apply a supervised contrastive loss on the incoming data.

$$\mathcal{L}_{SC}(\mathbf{X}) = -\sum_{\mathbf{x}_i \in \mathbf{X}} \frac{1}{|A(i)|} \mathcal{L}_{SC}(\mathbf{x}_i) \tag{1}$$

Where each sample's loss is given:

$$\mathcal{L}_{SC}(\mathbf{x}_i) = \\ \sum_{\mathbf{x}_p \in A(i)} \log \frac{h(g \circ f(\mathbf{x}_p), g \circ f(\mathbf{x}_i))}{\sum_{\mathbf{x}_a \in \mathbf{X}/x_i} h(g \circ f(\mathbf{x}_a), g \circ f(\mathbf{x}_i))} \tag{2}$$

Where $h(a, b) = \exp(\texttt{sim}(a, b)/\tau)$ and $\texttt{sim}(a, b) = \frac{a^T b}{\|a\|\|b\|}$. Here $A(i)$ represents the set of samples that form positive pairs with $x_i$ i.e. augmented views of $\mathbf{x}_i$ and other samples of the same class $\{\mathbf{x}_j | y_j = y_i\}$. Note that this loss is composed of tightness terms between positive pairs and contrast terms with negative pairs (Boudiaf et al., 2020).

**Prototype Learning without Contrasts** In order to easily link the discriminative representations learned by optimizing

$\mathcal{L}_{SC}(\mathbf{X})$ to a final class level prediction we consider the notion of class prototypes (Caccia et al., 2022; De Lange & Tuytelaars, 2021), which allow us to score a sample's representation with respect to each class. A simple solution for learning the class prototypes is to apply the Softmax in combination with the Cross-Entropy loss, for a given sample yielding

$$-\text{sim}(\mathbf{p}, f_\theta(\mathbf{x}_i)) + \log\Big( \sum_{\mathbf{p}_k \in \mathbf{P}} h(\mathbf{p}_k, f_\theta(\mathbf{x}_i)) \Big) \quad (3)$$

However, it has been shown that in the class-incremental setting, the softmax combined with cross-entropy produces a large interference with previously learned classes due to terms that suppress previous classes logits (Caccia et al., 2022; Ahn et al., 2021). Here, we propose instead to learn class prototypes that are representatives of each class samples' using only the first term in this loss, referred to as the "tightness" term (Boudiaf et al., 2020). For each class $c$ we initialize a random prototype $\mathbf{p}_c \in \mathbb{R}^d$. We want to optimize these prototypes to be representatives of current classes' samples without introducing any suppression to prototypes of previous classes. To achieve this we use a loss term considering only positive pairs of class samples and their corresponding prototypes where we aim to maximize the similarity of these pairs:

$$\mathcal{L}_p(\mathbf{X}) = -\frac{1}{|\mathbf{X}|} \sum_{\mathbf{x}_i, y_i \in \mathbf{X}, \mathbf{Y}} \text{sim}(\mathbf{p}_{y_i}, \text{sg}[f_\theta(\mathbf{x}_i)]) \quad (4)$$

Here $sg$ denotes the stop-gradient operations. The suggested loss contains only a tightness term, *i.e.*, contrast-free, which doesn't have a direct effect on previous classes prototypes. From this loss term, we aim to only optimize the prototypes and not to change the samples representations as this is taken care of by (1). Note that contrary to (Caccia et al., 2022) which also uses prototype-based learning, we do not include any contrastive terms for the prototype learning, the learning of class separations being left to $\mathcal{L}_{SC}$. Note that we utilize the stop-gradient operation so that the learning of the prototypes does not interfere with the representation learning or previous prototypes.

Once prototypes are obtained we can now directly perform predictions at test time by using the similarity of the sample representation and the set of prototypes to decide on the nearest class prototype.

**Prototypes-Samples Similarity Distillation**   Our prototypes are learned in isolation for each task. However, as we update our feature extractor using the supervised contrastive loss Eq. (1) prototypes of previous task classes will become outdated leading to the forgetting of previously learned classes. As shown in (Davari et al., 2022) this forgetting may correspond simply to movement in the decision boundary, despite classes still being well separated. To update old

prototypes as we update our representation, we propose a similarity distillation term using new class data as a proxy for old data. Before the start of a new training session, we compute the prototypes' similarities to each sample of new classes. During the new training session, we propose to minimize the KL divergence between the similarities distribution of prototypes to minibatch samples, enforcing current similarities to be similar to previous similarities.

Consider the current model and set of prototypes for previous classes $f_{\theta_t}, \mathbf{P}_o^t$ along with their corresponding model and prototype from the end of the previous task $f_{\theta_{t-1}}, \mathbf{P}_o^{t-1}$. For an incoming mini-batch $\mathbf{X}$ and a corresponding prototype we can consider the softmax output $\mathcal{P}_t(\mathbf{p}_k^t, \mathbf{X})$, where the $i^{th}$ entry is given:

$$\mathcal{P}_t(\mathbf{p}_k^t, \mathbf{X})_i = \frac{h(\mathbf{p}_k^t, f_{\theta_t}(\mathbf{x}_i))}{\sum_{\mathbf{x_j} \in \mathbf{X}} h(\mathbf{p}_k^t, f_{\theta_t}(\mathbf{x}_j))} \quad (5)$$

Denoting for shorthand $\mathcal{P}_t(\mathbf{p}_k^t, \mathbf{X})$ as $\mathcal{P}_t(k)$ we can now construct a relation distillation term as the KL-divergence between prototype-samples similarity distribution estimated with the model at session $t-1$ and during the current session $t$.

$$\mathcal{L}_d(\mathbf{P}) = \sum_{\mathbf{p}_k \in \mathbf{P}_o} KL\Big( \mathcal{P}_t(k) \,\|\, \mathcal{P}_{t-1}(k) \Big) \quad (6)$$

Note that this is distinct from distillation approaches where we compute the similarities for each sample over existing classes. As illustrated in Fig.1 the relative positions of samples to the prototypes are encouraged to remain the same by our loss. This results in flexibility in the representations in order to adapt to new classes while keeping the relative distances of many samples to the prototype as similar as possible. Our overall training objective is thus given as a combination of these three terms:

$$\mathcal{L}(\mathbf{X}) = \mathcal{L}_{sc}(\mathbf{X}) + \alpha \mathcal{L}_p(\mathbf{X}, P_c) + \beta \mathcal{L}_d(\mathbf{X}, P_o)$$

## 4. Experiments

In this section, we evaluate our proposed method on a wide range of challenging CL settings. In Sec. 4.1, we focus on the *task-incremental* (multi-head) setting, where we compare our method with other replay-free methods. Sec. 4.2 is dedicated to *class-incremental* setting, where the shared output layer poses an enormous challenge that drives most work to employ a replay buffer, unlike our replay-free solution. The evaluations are based on *average observed accuracy*. Specifically, we measure observed accuracy, $A_{ij}$, as the accuracy of the model after step $i$ on the test data of task $j$. Similarly, the average observed accuracy at the end of the sequence is $\frac{1}{T} \sum_{t \in T} A_{T,t}$ as used in (Li & Hoiem, 2017).

**Datasets**   In our experiments, we use Split-CIFAR100 (Krizhevsky et al., 2009), Split-MiniImageNet,
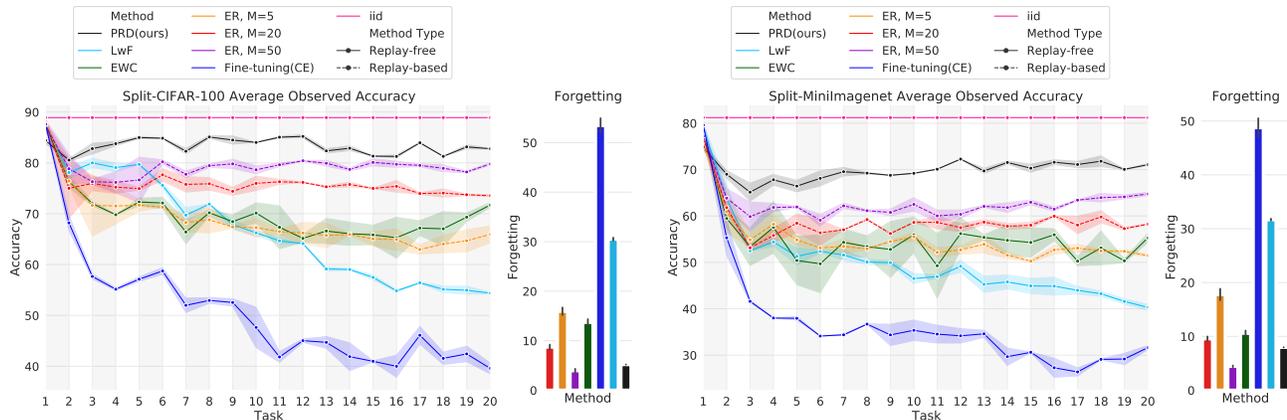
Figure 2. *Task-incremental* accuracy on 20-Task Split-CIFAR100(left) and Split-MiniImageNet(right). We observe that PRD widely outperforms other baselines without storing any previous task data, and as well exceeds the performance of ER with a very large buffer.

and ImageNet32 (Chrabaszcz et al., 2017; Davari et al., 2022) as the benchmarks for both multi-head and single-head settings. Split-CIFAR100 (Krizhevsky et al., 2009) comprises 20 tasks, each containing a disjoint set of 5 labels. The classes splits are constructed as in (Chaudhry et al., 2019). All CIFAR experiments process $32 \times 32$ images. Split-MiniImageNet divides the MiniImagenet dataset into 20 disjoint tasks of 5 labels each. Images are $84 \times 84$. ImageNet32 (Chrabaszcz et al., 2017) is a downsampled ($32 \times 32$ ) version of the entire ImageNet (Deng et al., 2009) dataset split into 200 tasks of 5 classes each. We use ImageNet32 in order to compare methods performance in very long sequences scenario.

**Baselines** Although our proposed method does not use any replay buffer, we consider in our evaluation both *replay-free* and *replay-based* methods, as replay-based have been shown to outperform other approaches in the continual learning setting (Chaudhry et al., 2019; Aljundi et al., 2019; Ji et al., 2020; Rebuffi et al., 2017). We consider the following replay-free methods in our evaluations:

**LwF** (Li & Hoiem, 2017): knowledge distillation based on current task data is used to limit forgetting.
**EWC** (Huszár, 2017): estimates an importance value for each parameter in the network and penalizes changes on parameters deemed important for previous tasks.
**SPB(Wu et al., 2021)**: A recent method that also utilizes contrastive learning and does not rely on replay data. We were unable to effectively reproduce their results since the code is not provided. However, we compare our approach directly to the reported results in the setting studied in the original work (Wu et al., 2021) in Sec. 4.2.
**iid**: The learner is trained on the whole data, in a single task containing all the classes.

The incorporated replay-based baselines are as follows:

**ER** (Chaudhry et al., 2019): Experience Replay with a buffer of a fixed size. In our experiments, we used buffer sizes of 5, 20, and 50 samples per class based on the evaluation setting. Note this is a very strong baseline that exceeds most methods, particularly with large buffers (50 samples) (Davari et al., 2022).
**iCaRL** (Rebuffi et al., 2017): A distillation loss alongside binary cross-entropy loss is used during training. Samples are classified based on the closest class prototypes.
**ER-AML** (Caccia et al., 2022): Utilizes SupCon loss, alongside a replay buffer, to reduce the representation drift of previously observed classes.
**ER-ACE** (Caccia et al., 2022): Similar to ER-AML, however, ER-ACE introduces a modified version of the standard softmax-crossentropy.

**Hyperparameter selection** For each method, optimal hyperparameters were selected via a grid search performed on the validation set. The selection process was done on a per-dataset basis, that is we picked the configuration which maximized the accuracy averaged over different settings. We found that for our method, the same hyperparameter configuration worked well across all settings and datasets. All necessary details to reproduce our experiments can be found in the supplementary materials.

### 4.1. Evaluations on Task-Incremental Setting

We evaluate Split-CIFAR100, Split-MiniImagenet, and ImageNet32 using the protocol from (Aljundi et al., 2019) with 100 training epochs training per task. We report the mean and standard error over 3 runs.

**Split-CIFAR100 and Split-MiniImageNet** We consider Split-CIFAR100 and Split-MiniImageNet with 20 tasks of 5 classes each. The results can be found in Figure 2, for Split-CIFAR100 and Split-MiniImageNet, using different buffer sizes for ER. In this setting, we can observe that our
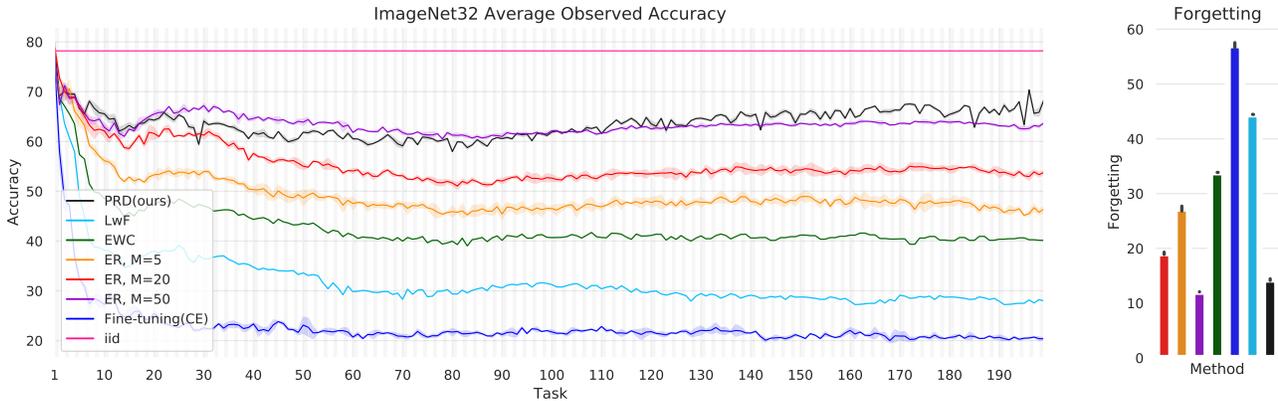
Figure 3. *Task-incremental* accuracy on 200-Task ImageNet32. On this long sequence, PRD matches a baseline with a large replay buffer. Other methods degrade over time, but the average accuracy of PRD improves due to the cumulative effect of maintaining better plasticity.
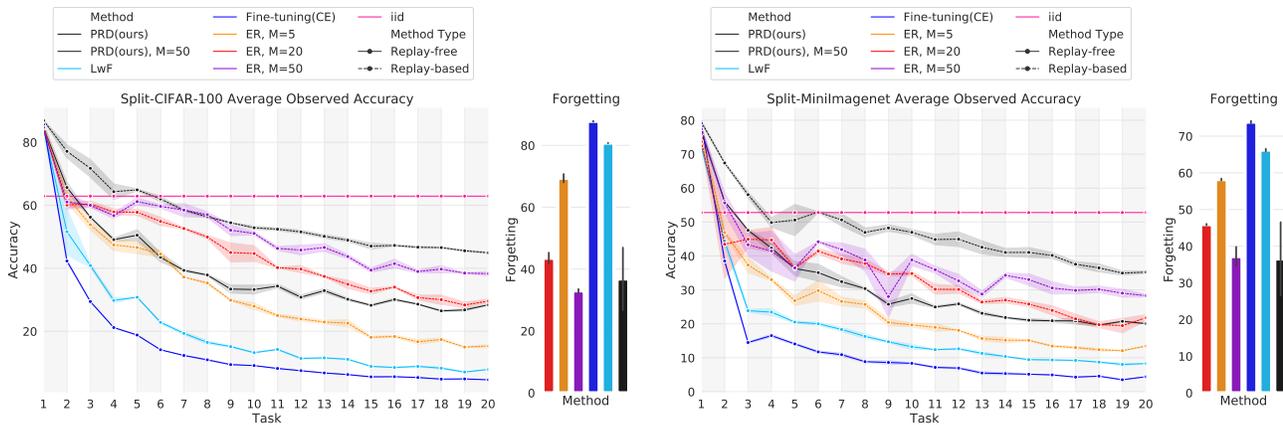


Figure 4. *Class-incremental* accuracy on 20-Task Split-CIFAR100(left) and Split-MiniImageNet(right). We observe that PRD outperforms not only other *replay-free* baselines but also ER, M=5, and is on par with ER, M=20, without storing any data. We also observe that with additional replay samples, PRD M=50 outperforms ER M=50 with the same number of replay samples.

proposed method *consistently outperforms* other methods by a significant margin. Even though our method does not utilize previous tasks' data in any form, it still outperforms ER with 50 replay samples per class, nearly closing the gap with the oracle *iid* setting. From Figure 2, we can observe that during the whole continual sequence of tasks, the average accuracy of our method on the observed classes remains relatively similar and even increases at several points during the sequence, *e.g.* $12^{th}$ task, suggesting a good trade-off between stability and plasticity of the model.

Although our method, in terms of average observed accuracy, outperforms other baselines with a considerable margin, a little higher forgetting rate can be observed compared to the strong replay-based baseline, *i.e.* ER with 50 replay samples. In Sec. 4.3, we show that our proposed method, in terms of plasticity, *i.e.* ability to learn new tasks, is comparable to naive fine-tuning, which is the upper bound among existing CL methods due to the absence of constraints on preserving previous tasks ( i.e., lacking stability). This

suggests that PRD is able to preserve previous tasks' information without losing the ability to learn new tasks.

**ImageNet32 - Long Task Sequence** We now consider a longer sequence than typically studied which allows us to observe whether the trends we have seen so far continue to hold. Using Imagenet32 we construct 200 tasks of 5 classes each. Figure 3 shows the average observed accuracy throughout the whole 200 tasks sequence. We see that in a very long sequence of tasks, the previously established observations about our method hold. Specifically, we see that as the model reaches the later stages of the sequence, our method outperforms the competitive baseline of ER with 50 replay samples *without* utilizing previous tasks' data in any form. Note that the number of stored data points leveraged by ER increases as we proceed in the sequence. Furthermore, we observe as in the previous section that the average observed accuracy of our proposed method not only stays relatively the same during the beginning but also

| Method | Split-CIFAR100, K = 20 | | | | Split-MiniImageNet, K = 20 | | | |
|---|---|---|---|---|---|---|---|---|
| | $M = 0$ | $M = 5$ | $M = 20$ | $M = 50$ | $M = 0$ | $M = 5$ | $M = 20$ | $M = 50$ |
| iid | 65.3 | 65.3 | 65.3 | 65.3 | 54.5 | 54.5 | 54.5 | 54.5 |
| Fine-tuning | 4.6 | 4.6 | 4.6 | 4.6 | 4.4 | 4.4 | 4.4 | 4.4 |
| ER (Chaudhry et al., 2019) | - | $15.2_{\pm0.7}$ | $29.6_{\pm0.8}$ | $38.3_{\pm0.9}$ | - | $13.4_{\pm0.2}$ | $21.7_{\pm0.8}$ | $28.7_{\pm0.5}$ |
| iCaRL (Rebuffi et al., 2017) | - | $19.8_{\pm0.5}$ | $28.6_{\pm0.7}$ | $32.9_{\pm0.5}$ | - | $16.2_{\pm0.1}$ | $22.8_{\pm0.3}$ | $26.1_{\pm0.2}$ |
| ER-AML (Caccia et al., 2022) | - | $21.4_{\pm0.8}$ | $35.3_{\pm0.6}$ | $42.4_{\pm0.8}$ | - | $17.1_{\pm0.3}$ | $26.3_{\pm0.7}$ | $32.3_{\pm0.2}$ |
| ER-ACE (Caccia et al., 2022) | - | $22.8_{\pm0.5}$ | $35.7_{\pm0.2}$ | $43.3_{\pm0.2}$ | - | $18.8_{\pm0.1}$ | $27.1_{\pm0.5}$ | $34.2_{\pm0.5}$ |
| PRD(Ours) | $\mathbf{27.8}_{\pm0.2}$ | $\mathbf{32.0}_{\pm0.4}$ | $\mathbf{39.5}_{\pm0.4}$ | $\mathbf{45.1}_{\pm0.5}$ | $\mathbf{20.0}_{\pm0.1}$ | $\mathbf{25.7}_{\pm0.3}$ | $\mathbf{31.3}_{\pm0.5}$ | $\mathbf{35.8}_{\pm0.4}$ |

*Table 1. Class-incremental* results on 20-Task Split-CIFAR100 and Split-MiniImageNet datasets using different buffer sizes. We observe that even with no replay samples (M=0) PRD outperforms all of the replay-based baselines with 5 replay samples. With a small number of replay samples, *e.g.* M=5, PRD widely outperforms other replay-based methods, suggesting the ability of our method to utilize replay samples while maintaining good performance with no access to prior data.

starts increasing as the model reaches the middle of the sequence, *i.e.* the $90^{th}$ task. This observation suggests that our method is able to efficiently learn new tasks' features while preserving the previous tasks' information.

### 4.2. Class-Incremental Setting

In addition to the experiments in the *task-incremental* setting, to further verify the effectiveness of our method in mitigating representation forgetting with no access to prior task data, we also evaluate on the more challenging *class-incremental* setting where we examine the ability to incrementally learn a shared classifier. Here as well we report the mean and standard error over 3 runs.

**Split-CIFAR100 and Split-MiniImageNet** Figure 4 shows the average observed *class-incremental* accuracy of the model over the 20 task sequence of Split-CIFAR100 and Split-MiniImageNet. Note that the *replay-based* methods are plotted in dashed lines. We can see that our method, with no access to previous tasks data, not only outperforms other *replay-free* methods but also beats ER with 5 replay samples and is on par with ER with 20 replay samples per class. From Figure 4, we can observe that the average accuracy of our method drops initially, probably due to the drift of the old tasks' prototypical features, but stays relatively the same from the middle of the sequence. This observation also suggests that in longer tasks sequences the learned prototypical features of old classes remain useful, even in the absence of any replay data.

In the following section, Sec. 4.2, we perform a thorough experiment on the effect of different replay buffer sizes, showing that our method beats the state-of-the-art replay-based methods with fewer stored samples.

**Leveraging stored samples** Our method targets incremental learning in scenarios where no stored samples are allowed. However, here, we investigate if our method can benefit from the availability of few stored samples. When utilizing replay data we follow the standard approach

| Method | Split-CIFAR100 | | ImageNet-Sub | |
|---|---|---|---|---|
| | K=6 | K=11 | K=6 | K=11 |
| iid | 73.4 | 73.2 | 82.0 | 82.7 |
| Fine-tuning | 22.3 | 12.6 | 23.6 | 13.2 |
| LwF-E (Yu et al., 2020) | 57.0 | 56.8 | 65.5 | 65.6 |
| EWC-E (Yu et al., 2020) | 56.3 | 55.4 | 65.2 | 64.1 |
| MAS-E (Yu et al., 2020) | 56.9 | 56.6 | 65.8 | 65.8 |
| SDC (Yu et al., 2020) | 57.1 | 56.8 | 65.6 | 65.7 |
| SPB (Wu et al., 2021) | 60.9 | 60.4 | 68.7 | 67.2 |
| PRD (Ours) | **64.3** | **63.7** | **71.8** | **70.3** |

*Table 2.* Pre-trained Initialization. We report average *cumulative* incremental accuracies over all tasks. PRD exceeds recent proposals in this challenging setting.

of ER-based methods, sampling half the training data of the mini-batch from the previous data. The subsequent optimization problem is kept the same. Note that now the relation distillation will also see data from past tasks that directly correspond to the prototypes. Tab. 1 compares our method with different buffer sizes to other replay-based methods. It can be seen that our method can successfully leverage the available data and further improve the performance achieving high gains over state of art in low buffer regime. This suggests our method is highly effective in both limited and no replay data settings.

**Pre-trained Initialization** To further measure the class-incremental performance of our method and allow direct comparison to (Wu et al., 2021), we also evaluate our method on Split-CIFAR100 (Krizhevsky et al., 2009) and ImageNet-Subset (Rebuffi et al., 2017; Krizhevsky et al., 2012) using the protocol and constraints from (Wu et al., 2021). In these settings half the classes are used for an initial pre-training phase. ImageNet-Subset contains 100 classes randomly sampled from ImageNet. Following (Wu et al., 2021), we randomly select the first 50 classes as the 1-*st* phase and evenly split the remaining 50 classes for *K*-1 phases. Similar to (Wu et al., 2021), for this experiment, we evaluate our models with *K*=6 and 11 phases on both
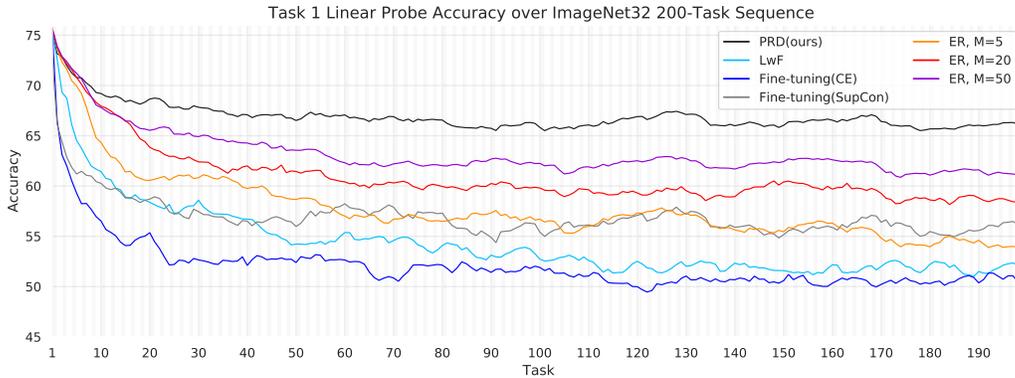
*Figure 5.* Task 1 LP accuracy over 200-Task ImageNet32. We compare Linear probe accuracy for Tasks 1 data over the whole sequence. We can observe that during a long sequence, the performance of our method, *i.e.* PRD, not only stays relatively flat, but also increases at some points in the later stages of the sequence, suggesting its ability to preserve the information of observed tasks.
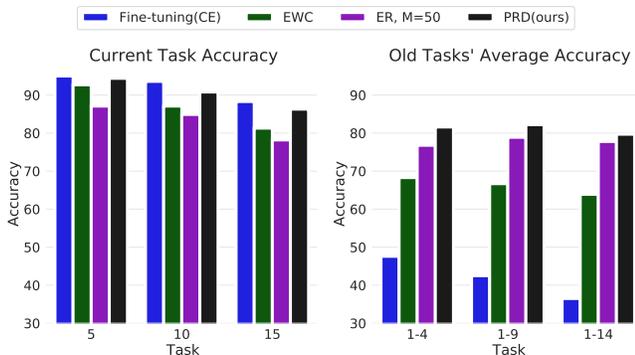


*Figure 6. Task-incremental* Split-CIFAR100. Accuracy on the current task(left) and average accuracy on previous tasks(right). PRD performs well on the current task while having low forgetting.

Split-CIFAR100 and ImageNet-Subset datasets, *i.e.*, after the $1^{st}$ phase, we incrementally add 5 or 10 new classes at each phase. Following (Wu et al., 2021), we report the average *cumulative* incremental accuracy over all phases. All results are averaged over three runs.

Tab. 2 shows average *cumulative* incremental accuracies (as used in (Wu et al., 2021)) over all phases on Split-CIFAR100 and ImageNet-Subset. We observe that our method exceeds the recent proposal of (Wu et al., 2021) in this setting as well as beating strong baselines such as SDC. Note that (Wu et al., 2021) also applied a self-supervised objective which we do not include as we were unable to obtain source code for these experiments, and this was a complementary approach that can enhance our method as well.

### 4.3. Analysis and Ablations

**PRD Balances Plasticity and Stability** A continual learner should be able to easily integrate new knowledge from new tasks (plasticity) while benefiting from prior knowledge to improve performance on the current task

(forward transfer). Continual learning methods are often characterized by a trade-off in plasticity and stability. Stability refers to the ability to retain the knowledge of prior tasks(Mermillod et al., 2013), often measured by the forgetting metric. We have thus far shown that PRD has relatively low forgetting, for example in Fig. 2, for CIFAR-100 it has the lowest forgetting, only slightly improved on by the ER-M50 a baseline with large replay buffer. Both ER-M50 and PRD have good stability, but their plasticity can be difficult to gauge in long task sequences directly from observed accuracy. For example, a task can be very poorly learned during a session but learned later on thanks to the replay buffer. We thus directly compare the current task performance separately from the old task performance corresponding to PRD, ER M=50, and EWC on task-incremental CIFAR100, corresponding to the results in Fig. 2. The results are shown for tasks 5,10, and 15 in Fig. 6.

We first observe that all methods progressively degrade in current task accuracy. Since tasks are sampled uniformly from the set of possible tasks we can assume this corresponds to a gradual reduction in plasticity. This is consistent with many previous observations of continual learning systems(Dohare et al., 2021; Mermillod et al., 2013). On the other hand, we observe that compared to other strong baselines like EWC and ER, M=50, the current task accuracy of PRD is substantially higher, while the old task accuracy is as well, being largely maintained as training progresses. Thus PRD provides a strong tradeoff in plasticity and stability. If we observe the behavior of ER, M=50, we see that its old task accuracy is sometimes increasing (for example at task 10). Overall, we can state that models trained by our method, PRD, exhibit a plasticity close to the constraint free fine-tuning while showing the best stability.

**Representation Forgetting** Following (Davari et al., 2022), we evaluate the representation forgetting of our method against other baselines with a *Linear Probe*, de-

noted as LP. Similar to the definition of observed accuracy in Sec. 4, we can measure the LP accuracy for each step $i$ and task $j$ as well as the average LP accuracy.

Similar to (Davari et al., 2022), we construct 200 tasks of 5 classes using ImageNet32 dataset in the task-incremental setting. Figure 5 shows the performance of the model on the first task throughout the whole continual sequence. We observe that although the naive SupCon exceeds replay with M=5 in terms of representation forgetting on the first task, PRD provides a very substantial improvement. This suggests that we not only benefit from stabilizing the prototypes but the representation itself greatly benefits from PRD, avoiding forgetting at the representation level.

## 5. Conclusion

We proposed a novel approach for Replay-Free Continual learning that effectively leverages relationship distillation alongside supervised contrastive learning. On a wide array of evaluations, our method is shown to provide good trade-offs in stability and plasticity, leading to large improvements over replay-free baselines and allows us to exceed performance of replay-based methods. Moreover, we showed that our method can effectively utilize additional replay samples, outperforming the state-of-the-art replay-based methods in the class-incremental setting. These observations open up potential new directions for approaches in replay-free continual learning.

## Acknowledgements

## References

Ahn, H., Kwak, J., Lim, S., Bang, H., Kim, H., and Moon, T. Ss-il: Separated softmax for incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 844–853, October 2021.

Aljundi, R., Chakravarty, P., and Tuytelaars, T. Expert gate: Lifelong learning with a network of experts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Aljundi, R., Caccia, L., Belilovsky, E., Caccia, M., Charlin, L., and Tuytelaars, T. Online continual learning with maximally interfered retrieval. In *Advances in Neural Information Processing (NeurIPS)*, 2019.

Asadi, N., Mudur, S., and Belilovsky, E. Tackling online one-class incremental learning by removing negative contrasts. *arXiv preprint arXiv:2203.13307*, 2022.

Barletti, T., Biondi, N., Pernici, F., Bruni, M., and Del Bimbo, A. Contrastive supervised distillation for continual representation learning. In *International Conference on Image Analysis and Processing*, pp. 597–609. Springer, 2022.

Boudiaf, M., Rony, J., Ziko, I. M., Granger, E., Pedersoli, M., Piantanida, P., and Ayed, I. B. A unifying mutual information view of metric learning: cross-entropy vs. pairwise losses. In *European conference on computer vision*, pp. 548–564. Springer, 2020.

Caccia, L., Aljundi, R., Asadi, N., Tuytelaars, T., Pineau, J., and Belilovsky, E. New insights on reducing abrupt representation change in online continual learning. *arXiv preprint arXiv:2203.03798*, 2022.

Caccia, M., Rodriguez, P., Ostapenko, O., Normandin, F., Lin, M., Caccia, L., Laradji, I., Rish, I., Lacoste, A., Vazquez, D., et al. Online fast adaptation and knowledge accumulation: a new approach to continual learning. *arXiv preprint arXiv:2003.05856*, 2020.

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

Chaudhry, A., Ranzato, M., Rohrbach, M., and Elhoseiny, M. Efficient lifelong learning with a-gem. In *ICLR 2019*.

Chaudhry, A., Dokania, P. K., Ajanthan, T., and Torr, P. H. Riemannian walk for incremental learning: Understanding forgetting and intransigence. *arXiv preprint arXiv:1801.10112*, 2018.

Chaudhry, A., Rohrbach, M., Elhoseiny, M., Ajanthan, T., Dokania, P. K., Torr, P. H., and Ranzato, M. Continual learning with tiny episodic memories. *arXiv preprint arXiv:1902.10486*, 2019.

Chrabaszcz, P., Loshchilov, I., and Hutter, F. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017.

Davari, M., Asadi, N., Mudur, S., Aljundi, R., and Belilovsky, E. Probing representation forgetting in supervised and unsupervised continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16712–16721, 2022.

De Lange, M. and Tuytelaars, T. Continual prototype evolution: Learning online from non-stationary data streams. *arXiv preprint arXiv:2009.00919*, 2020.

De Lange, M. and Tuytelaars, T. Continual prototype evolution: Learning online from non-stationary data streams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8250–8259, 2021.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Dohare, S., Mahmood, A. R., and Sutton, R. S. Continual backprop: Stochastic gradient descent with persistent randomness. *arXiv preprint arXiv:2108.06325*, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Huszár, F. On quadratic penalties in elastic weight consolidation. *arXiv preprint arXiv:1712.03847*, 2017.

Javed, K. and Shafait, F. Revisiting distillation and incremental classifier learning. In *Asian conference on computer vision*, pp. 3–17. Springer, 2018.

Ji, X., Henriques, J., Tuytelaars, T., and Vedaldi, A. Automatic recall machines: Internal replay, continual learning and the brain. *arXiv preprint arXiv:2006.12323*, 2020.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *arXiv preprint arXiv:1612.00796*, 2016.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Li, X., Zhou, Y., Wu, T., Socher, R., and Xiong, C. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *International Conference on Machine Learning*, pp. 3925–3934. PMLR, 2019.

Li, Z. and Hoiem, D. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017.

Lopez-Paz, D. et al. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pp. 6467–6476, 2017.

McCloskey, M. and Cohen, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of learning and motivation*, 24:109–165, 1989.

Mermillod, M., Bugaiska, A., and Bonin, P. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects, 2013.

Myung, I. J. Tutorial on maximum likelihood estimation. *Journal of mathematical Psychology*, 47(1):90–100, 2003.

Park, W., Kim, D., Lu, Y., and Cho, M. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3967–3976, 2019.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. icarl: Incremental classifier and representation learning. In *Proc. CVPR*, 2017.

Rosenfeld, A. and Tsotsos, J. K. Incremental learning through deep adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 42(3):651–663, 2018.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Tian, Y., Krishnan, D., and Isola, P. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019.

Verwimp, E., Yang, K., Parisot, S., Lanqing, H., McDonagh, S., Pérez-Pellitero, E., De Lange, M., and Tuytelaars, T. Clad: A realistic continual learning benchmark for autonomous driving. *arXiv preprint arXiv:2210.03482*, 2022.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

Wu, G., Gong, S., and Li, P. Striking a balance between stability and plasticity for class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1124–1133, 2021.

Yu, L., Twardowski, B., Liu, X., Herranz, L., Wang, K., Cheng, Y., Jui, S., and Weijer, J. v. d. Semantic drift compensation for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6982–6991, 2020.

Zhu, F., Zhang, X.-Y., Wang, C., Yin, F., and Liu, C.-L. Prototype augmentation and self-supervision for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5871–5880, 2021a.

Zhu, J., Tang, S., Chen, D., Yu, S., Liu, Y., Rong, M., Yang, A., and Wang, X. Complementary relation contrastive distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9260–9269, 2021b.

# APPENDIX

## A. Experimental Setup

In this section, we provide additional details regarding the baselines and hyperparameters. In all experiments, we leave the **batch size** and the **number of epochs** fixed at **128** and **100**. The model architecture ($\theta$) is also kept constant, which is a regular ResNet-18 model, where the dimensions of the last linear layer change depending on the input height and width.

The augmentation pipeline is consistent across all experiments, consisting of random crop, random horizontal flip, color jitter of 0.4, and random grayscale.

**Hyperparameter Selection**    All results in the paper have been either implemented by us or adapted from (Caccia et al., 2022), with the exception of SPB (Wu et al., 2021), where results were taken from the original paper since there was no public codebase for that baseline at the time of submission. For each method, a grid search was run on the possible hparams, which we detail below. In the following, we list the hyperparameters that we included in our grid search. The best values for each parameter are underlined.

**PRD(ours)**

- LR: [0.01, 0.005, 0.001]
- SupCon Temperature: [0.1, 0.2, 0.3]
- Relation Distillation Coefficient($\beta$): [1., 2., 4., 8., 16.]
- Prototypes Coefficient($\alpha$): [1., 2., 4.]

**EWC** (Huszár, 2017):

- LR: [0.01, 0.005, 0.001]
- Lambda Coefficient: [20, 50, 100, 200, 500, 1000]

**LwF** (Li & Hoiem, 2017), **ER** (Chaudhry et al., 2019), and **iCaRL** (Rebuffi et al., 2017):

- LR: [0.01, 0.005, 0.001]

Similar to (Caccia et al., 2022), for ER, rehearsal begins as soon as the buffer is not empty. Also when samples are being fetched from the buffer, we do not exclude classes from the current task.

**ER-ACE** (Caccia et al., 2022):

- LR: [0.01, 0.005, 0.001]

Following (Caccia et al., 2022) implementation, for the masking loss, we simply use `logits.maskedfill(mask, -1e9)` to filter out classes which should not receive gradient.

**ER-AML** (Caccia et al., 2022):

- LR: [0.01, 0.005, 0.001]
- SupCon Temperature: [0.1, 0.2, 0.3]

## B. Ablation on Prototypes-Samples Similarity Distillation

As discussed in Sec. 4.3, continual learning methods are characterized by a trade-off in plasticity and stability. Stability refers to the ability to retain the knowledge of prior tasks(Mermillod et al., 2013), often measured by the forgetting metric. According to our observations from Sec. 4.3, one can observe the PRD has relatively low forgetting while maintaining high

plasticity in learning new tasks. PRD controls the stability-plasticity trade-off mostly using the coefficient for *prototype-sample relation distillation* loss. Here we do an ablation on the effect of our prototype-sample relation distillation loss in three datasets, Split-CIFAR100, Split-MiniImageNet, and ImageNet32. Tab. 3 presents the performance of our method with different *coefficient* values($\beta$) for our prototype-sample relation distillation loss.

We can observe that using a coefficient value of 0 ($\beta = 0$), *i.e.* having no relation distillation loss, Eq. (3), results in very low average accuracy for all of the three datasets. This observation shows the importance of relation distillation loss in remembering old tasks' information. Further, we can observe using different values of $\beta$, with shorter task sequences like 20 task Split-CIFAR100, does not affect the overall average performance of the model across all tasks. On the other hand, with long task sequences, *e.g.* 200 task ImageNet32, a higher coefficient value for the distillation loss results in less forgetting and better overall average accuracy.

| Distillation Coefficient($\beta$) | Dataset | | |
| | Split-CIFAR100 (K=20) | Split-MiniImageNet (K=20) | ImageNet32 (K=200) |
| --- | --- | --- | --- |
| $\beta = 0.$ | $39.4_{\pm 1.5}$ | $31.2_{\pm 0.9}$ | $21.3_{\pm 0.8}$ |
| $\beta = 1.$ | $80.0_{\pm 0.5}$ | $59.3_{\pm 0.3}$ | $55.4_{\pm 0.4}$ |
| $\beta = 2.$ | $82.1_{\pm 0.3}$ | $63.7_{\pm 0.3}$ | $59.2_{\pm 0.4}$ |
| $\beta = 4.$ | $\mathbf{83.5}_{\pm 0.4}$ | $\underline{68.3}_{\pm 0.4}$ | $62.7_{\pm 0.2}$ |
| $\beta = 8.$ | $\underline{83.1}_{\pm 0.4}$ | $\mathbf{70.9}_{\pm 0.5}$ | $\underline{65.1}_{\pm 0.3}$ |
| $\beta = 16.$ | $82.7_{\pm 0.2}$ | $67.2_{\pm 0.5}$ | $\mathbf{67.5}_{\pm 0.2}$ |

*Table 3.* Ablation study on the effect of relation distillation coefficient, $\beta$ in Eq. (3). The reported numbers are *Task-incremental* average observed accuracy. We can observe that $\beta = 0$, having no distillation, results in very low average accuracy over all tasks. On the other hand, when the sequence is very long, *e.g.* 200 task ImageNet32, a higher coefficient value for the distillation loss results in better overall average accuracy.

## C. Ablation on Prototype Learning without Contrasts

PRD uses class prototypes to score a sample's representation with respect to each class. Tab. 4 presents the performance of our method with different *coefficient values($\alpha$)* for prototypes learning loss ($\mathcal{L}_p$). When the corresponding coefficient value is set to 0 ($\alpha = 0$), which means no optimization for class prototypes in Eq. (3), the average accuracy across all three datasets is basically random. When different values of $\alpha$ are used, there is no significant effect on the overall average performance of the model for both shorter and longer task sequences.

| Prototypes Coefficient($\alpha$) | Dataset | |
| | Split-CIFAR100 (K=20) | Split-MiniImageNet (K=20) |
| --- | --- | --- |
| $\alpha = 0.$ | $20.6_{\pm 0.2}$ | $20.2_{\pm 0.2}$ |
| $\alpha = 1.$ | $81.8_{\pm 0.5}$ | $68.0_{\pm 0.4}$ |
| $\alpha = 2.$ | $82.2_{\pm 0.4}$ | $\mathbf{69.8}_{\pm 0.3}$ |
| $\alpha = 4.$ | $\mathbf{83.5}_{\pm 0.4}$ | $\underline{68.3}_{\pm 0.5}$ |
| $\alpha = 8.$ | $\underline{82.7}_{\pm 0.5}$ | $67.9_{\pm 0.3}$ |
| $\alpha = 16.$ | $82.3_{\pm 0.5}$ | $67.4_{\pm 0.4}$ |

*Table 4.* Ablation study on the effect of prototype learning coefficient, $\alpha$ in Eq. (3). The reported numbers are *Task-incremental* average observed accuracy.

## D. Domain-Incremental Experiment

We evaluate our approach using the CLAD-C dataset (Verwimp et al., 2022), under the conditions laid out in (Verwimp et al., 2022). The dataset contains images recorded via dashcams over 3 days. The shift between night and day constitutes the task boundaries, hence overall we have 6 tasks. The objective of each task is to correctly classify an image into one of 6 possible classes of objects: 1. pedestrian 2. cyclist 3. car 4. truck 5. bus 6. tricycle. The dataset reflects the real-world distribution of these objects. Hence, certain classes are rarely observed (e.g. the tricycle class) and others are seen more frequently (e.g. the car class). As the night and day changes in the data stream and we are introduced to new tasks, the

image distribution changes, sometimes so drastic that leads to the absence of a few classes. This fact, along with the in-task class imbalance of the data makes the CLAD-C dataset (Verwimp et al., 2022) a challenging, yet realistic, benchmark.

The training data contains overall 22,249 objects distributed over 6 tasks. We report our results using the final Average Mean Class Accuracy (AMCA) on the test data which contains 69,881 objects spanning both day and night. The AMCA for $T$ tasks each containing $C$ classes is given by:

$$\text{AMCA} = \frac{1}{|T||C|} \sum_{t \in T} \sum_{c \in C} A_c^t \tag{7}$$

where $A_c^t$ is the accuracy of the class $c$ for the task $t$. The results are given in Table 5. All methods in Table 5 use a ResNet-50 (He et al., 2016) architecture, pretrained on ImageNet (Deng et al., 2009), and use a batch size of 32. Our results highlight the versatility of our method and its applicability to real-life continual learning scenarios. Moreover, it suggests that our method is cable of performing under severe class imbalance and drastic distribution shifts, without having access to past data.

|      | Finetune | EWC (Kirkpatrick et al., 2016) | LwF (Li & Hoiem, 2017) | PRD (ours) |
|------|----------|--------------------------------|------------------------|------------|
| AMCA | 40.5     | 62.5                           | 63.7                   | **65.1**   |

*Table 5.* Domain-incremental setting using the CLAD-C dataset (Verwimp et al., 2022). All methods use a ResNet-50 (He et al., 2016) architecture, pre-trained on ImageNet (Deng et al., 2009), and use a batch size of 32. Our results highlight the versatility of our method and its applicability to real-life continual learning scenarios. Moreover, it suggests that our method is cable of performing under severe class imbalance and drastic distribution shifts, without having access to past data.