Regularizing with Pseudo-Negatives for Continual Self-Supervised Learning

Sungmin Cha[1] **Kyunghyun Cho**[1 2] **Taesup Moon**[3]

Abstract

We introduce a novel Pseudo-Negative Regularization (PNR) framework for effective continual self-supervised learning (CSSL). Our PNR leverages pseudo-negatives obtained through model-based augmentation in a way that newly learned representations may not contradict what has been learned in the past. Specifically, for the InfoNCE-based contrastive learning methods, we define symmetric pseudo-negatives obtained from current and previous models and use them in both main and regularization loss terms. Furthermore, we extend this idea to non-contrastive learning methods which do not inherently rely on negatives. For these methods, a pseudo-negative is defined as the output from the previous model for a differently augmented version of the anchor sample and is asymmetrically applied to the regularization term. Extensive experimental results demonstrate that our PNR framework achieves state-of-the-art performance in representation learning during CSSL by effectively balancing the trade-off between plasticity and stability.

1. Introduction

Self-Supervised Learning (SSL) has recently emerged as a cost-efficient approach for training neural networks, eliminating the need for laborious data labelling (Gui et al., 2023). Specifically, the representations learned by recent SSL methods (*e.g.,* MoCo (He et al., 2020), SimCLR (Chen et al., 2020a), BarlowTwins (Zbontar et al., 2021), BYOL (Grill et al., 2020), and VICReg (Bardes et al., 2022)) are shown to have excellent quality, comparable to those learned from supervised learning. Despite such success, huge memory and computational complexities are the apparent bottlenecks for easily maintaining and updating the self-supervised learned models, since they typically require large-scale unsupervised data, large mini-batch sizes, and numerous gradient update steps for training.

To that end, Continual Self-Supervised Learning (CSSL), in which the aim is to learn progressively improved representations from a sequence of unsupervised data, can be an efficient alternative to the high-cost, jointly trained self-supervised learning. With such motivation, several recent studies (Madaan et al., 2022; Hu et al., 2022; Fini et al., 2022) have considered the CSSL using various SSL methods and showed their effectiveness in maintaining representation continuity. Despite the positive results, we note that the core idea for those methods is mainly borrowed from the large body of continual learning research for supervised learning (Parisi et al., 2019; Delange et al., 2021; Wang et al., 2024). Namely, a typical supervised continual learning method can be generally described as employing a single-task loss term for the new task (*e.g.,* cross-entropy or supervised contrastive loss (Khosla et al., 2020)) together with a certain type of regularization (*e.g.,* distillation-based (Li & Hoiem, 2017; Douillard et al., 2020; Kang et al., 2022; Wang et al., 2022; Cha et al., 2021a) or norm-based (Kirkpatrick et al., 2017; Aljundi et al., 2018; Jung et al., 2020; Ahn et al., 2019; Cha et al., 2021b) or replay-sample based terms (Wu et al., 2019; Rebuffi et al., 2017)) to prevent forgetting; the recent state-of-the-art CSSL methods simply follow that approach with *self-supervised* loss terms (*e.g.*, CaSSLe (Fini et al., 2022)).

In this regard, we raise an issue on the current CSSL approach; the efficacy of simply incorporating a regularization term into the existing self-supervised loss for achieving successful CSSL remains uncertain. Namely, typical regularization terms are essentially designed to maintain the representations of the previous model, but they may hinder the capability of learning better representations while learning from the new task (*i.e., plasticity)* (Cha et al., 2024; Kim & Han, 2023).

To address these limitations, we propose a novel method called Pseudo-Negative Regularization (PNR) for CSSL, which utilizes *pseudo-negatives* for each anchor of a given input, obtained by model-based augmentation. In the CSSL using various SSL methods, PNR defines different pseudo-negatives tailored for each contrastive and non-contrastive

[1]New York University [2]Genentech [3]ASRI / INMC / IPAI / AIIS, Seoul National University. Correspondence to: Taesup Moon <tsmoon@snu.ac.kr>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

learning method, respectively. Firstly, we consider the case of using InfoNCE-type contrastive loss (Oord et al., 2018), such as SimCLR and MoCo, which explicitly leverages negative samples. We propose novel loss functions by modifying the ordinary InfoNCE loss and contrastive distillation (Tian et al., 2019), ensuring that the former considers negatives from the previous model and the latter includes negatives from the current model as the pseudo-negatives. As a result, these loss functions ensure that newly acquired representations do not overlap with previously learned ones, enhancing *plasticity*. Moreover, they enable effective distillation of prior knowledge into the current model without interfering with the representations already learned by the current model, thereby improving overall *stability*. Second, we extend the idea of using pseudo-negatives to CSSL using non-contrastive learning methods like BYOL, BarlowTwins, and VICReg, which do not explicitly employ negative samples in their original implementations. For this, we reevaluate the relationship between the anchor of the current model and the negatives from the previous model observed in the contrastive distillation Building on this relationship, we propose a novel regularization that defines the pseudo-negative for the anchor from the current model as the output feature of the same image with different augmentations from the previous model. Our final loss function aims to minimize their similarity by incorporating this regularization alongside the existing distillation term for CSSL. Finally, through extensive experiments, our proposed method not only achieves state-of-the-art performance in CSSL scenarios and downstream tasks but also shows both better *stability* and *plasticity*.

2. Related Work

Self-supervised representation learning There have been several recent variations for Self-Supervised Learning (SSL) (Alexey et al., 2016; Doersch et al., 2015; Vincent et al., 2010; Zhang et al., 2016; Hadsell et al., 2006; Chen et al., 2020a; He et al., 2020). Among those, *contrastive loss-based* methods have emerged as one of the leading approaches to learn discriminative representations (Hadsell et al., 2006; Oord et al., 2018), in which the representations are learned by pulling the positive pairs together and pushing the negative samples apart. Several efficient contrastive learning methods, like MoCo (He et al., 2020; Chen et al., 2020b), SimCLR (Chen et al., 2020a), have been proposed build on the InfoNCE loss (Oord et al., 2018). Additionally, *non-contrastive learning* methods, such as Barlow Twins (Zbontar et al., 2021), BYOL (Grill et al., 2020) and VICReg (Bardes et al., 2022), have been demonstrated to yield high-quality learned representations without using negative samples.

Continual learning Continual learning (CL) is the process of acquiring new knowledge while retaining previously learned knowledge (Parisi et al., 2019; Masana et al., 2022) from a sequence of tasks. To balance the trade-off between *plasticity*, the ability to learn new tasks well, and *stability*, the ability to retain knowledge of previous tasks (Mermillod et al., 2013), the supervised CL research has been proposed in several categories. For more details, one can refer to (Wang et al., 2024; Delange et al., 2021).

Continual Self-Supervised Learning Recently, there has been a growing interest in Continual Self-Supervised Learning (CSSL), as evidenced by several related researches (Rao et al., 2019; Madaan et al., 2022; Hu et al., 2022; Fini et al., 2022). While all of them explore the possibility of using unsupervised datasets for CL, they differ in their perspectives. (Rao et al., 2019) is the first to introduce the concept of unsupervised continual learning and proposed a novel approach to learning class-discriminative representations without any knowledge of task identity. (Madaan et al., 2022) proposes a novel data augmentation method for CSSL and first demonstrates that CSSL can outperform supervised CL algorithms in the task-incremental learning scenario. Another study (Hu et al., 2022) focuses on the benefits of CSSL in large-scale datasets (*e.g.*, ImageNet), demonstrating that a competitive pre-trained model can be obtained through CSSL. The first significant regularization for CSSL was proposed by CaSSLe (Fini et al., 2022). They devised a novel regularization that helps to overcome catastrophic forgetting in CSSL, achieving state-of-the-art performance in various scenarios without using exemplar memory. After that, several papers have been published but they consider settings that are different from CaSSLe. C^{2}ASR (Cheng et al., 2023) considers to use the exemplar memory and introduces both a novel loss function and exemplar sampling strategy. (Yu et al., 2024; Tang et al., 2024; Gomez-Villa et al., 2024) are tailored for semi-supervised learning scenarios and demonstrate superior performance in such cases. Additionally, (Yu et al., 2024; Gomez-Villa et al., 2024) are dynamic architecture-based algorithms where the model expands as the number of tasks grows.

In this paper, we offer a few distinctive contributions compared to above mentioned related works. First, we identify shortcomings in the conventional regularization-based loss formulation for CSSL, such as CaSSLe. Second, we introduce a novel concept of pseudo-negatives and propose a new loss function that incorporates this concept, applicable to both contrastive and non-contrastive learning-based CSSL.

3. Problem Setting

Notations and preliminaries. We evaluate the quality of CSSL methods using the setting and data as in (Fini et al., 2022). Namely, let t be the task index, where $t \in \{1, \ldots, T\}$, and T represent the maximum number of tasks. The input data and their corresponding true labels given at the t-th task are denoted by $\boldsymbol{x} \in \mathcal{X}_t$ and $y \in \mathcal{Y}_t$, respectively[1]. Let $\mathcal{D}$ is the entire dataset. We assume each training dataset for task t comprises M supervised pairs, denoted as $\mathcal{D}_t = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{M}$, in which each pair is considered to be sampled from a joint distribution $p(\mathcal{X}_t, \mathcal{Y}_t)$. Note in the case of continual supervised learning (CSL) (Delange et al., 2021; Masana et al., 2022), both inputs and the labels are used, whereas in CSSL (Fini et al., 2022; Madaan et al., 2022), only input data are utilized for training, while the true labels are used only for the evaluation of the learned representations, such as linear probing or k-NN evaluation (Fini et al., 2022; Cha et al., 2024). Let $m_{\boldsymbol{\psi}_t} \circ h_{\boldsymbol{\theta}_t}$ is the model consisting of the representation encoder (with parameter $\boldsymbol{\theta}_t$) and an MLP layer (with parameter $\boldsymbol{\psi}_t$) learned after task t. To evaluate the quality of $h_{\boldsymbol{\theta}_t}$ via linear probing, we consider a classifier $f_{\boldsymbol{\Theta}_t} = o_{\boldsymbol{\phi}_t} \circ h_{\boldsymbol{\theta}_t}$, in which $\boldsymbol{\Theta}_t = (\boldsymbol{\theta}_t, \boldsymbol{\phi}_t)$ and $o_{\boldsymbol{\phi}_t}$ is the linear output layer (with parameter $\boldsymbol{\phi}_t$) on top of $h_{\boldsymbol{\theta}_t}$. Then, only $o_{\boldsymbol{\phi}_t}$ is supervised trained (with frozen $h_{\boldsymbol{\theta}_t}$) using all the training dataset $\mathcal{D}_{1:t}$, including the labels, and the accuracy of resulting $f_{\boldsymbol{\Theta}_t}$ becomes the proxy for the representation quality.

Class-/Data-/Domain-incremental learning. We consider the three scenarios of continual learning as outlined in (Van de Ven & Tolias, 2019; Wang et al., 2024; Fini et al., 2022). We use k and j to denote arbitrary task numbers, where $k, j \in \{1, \ldots, T\}$ and $k \neq j$. The first category is the *class-incremental learning* (Class-IL), in which the t-th task's dataset consists of a unique set of classes for the input data, namely, $p(\mathcal{X}_k) \neq p(\mathcal{X}_j)$ and $\mathcal{Y}_k \cap \mathcal{Y}_j = \varnothing$. The second category is *domain-incremental learning* (Domain-IL), in which each dataset $\mathcal{D}_t$ has the same set of true labels but with different distribution on $\mathcal{X}_t$, denoted as $p(\mathcal{X}_k) \neq p(\mathcal{X}_j)$ but $\mathcal{Y}_k = \mathcal{Y}_j$. In other words, each dataset in Domain-IL contains input images sampled from a different domain, but the corresponding set of true labels is the same as for other tasks. Finally, we consider *data-incremental learning* (Data-IL), in which a set of input images $\mathcal{X}_t$ is sampled from a single distribution, $p(\mathcal{X}_k) = p(\mathcal{X}_j)$, but $\mathcal{Y}_k = \mathcal{Y}_j$. To implement the Data-IL scenario in our experiments, we shuffle the entire dataset (such as ImageNet-100) and divide it into T disjoint datasets.

4. Pseudo-Negative Regularization (PNR)

4.1. Motivation

Several studies have been conducted with the aim of progressively improving the quality of representations learned by the encoder ($h_{\boldsymbol{\theta}_t}$) in CSSL. While it has been noted that simple fine-tuning using D_t results in less severe forgetting compared to supervised continual learning (Madaan et al., 2022; Davari et al., 2022), CaSSLe (Fini et al., 2022) has achieved even more successful CSSL. It introduces a novel regularization method based on existing SSL methods, using the output features of both the encoder at time t and the encoder at time $t-1$, to overcome catastrophic forgetting in learned representations. Despite CaSSLe achieving promising results in various experiments, our motivation arises from the belief that incorporating *pseudo-negatives* can lead to even more successful results. Consider an augmented image obtained by applying different augmentations to the input image $\boldsymbol{x}$, denoted by $\boldsymbol{x}^A$ and $\boldsymbol{x}^B$. The output features of the $m_{\boldsymbol{\psi}_t} \circ h_{\boldsymbol{\theta}_t}$ and $m_{\boldsymbol{\psi}_{t-1}} \circ h_{\boldsymbol{\theta}_t-1}$ for these augmented images are denoted by $\boldsymbol{z}_t^A$, $\boldsymbol{z}_t^B$, $\boldsymbol{z}_{t-1}^A$, and $\boldsymbol{z}_{t-1}^B$, respectively. In contrast to CaSSLe, our approach involves utilizing the output features of both the models as pseudo-negatives, as illustrated in Figure 1. For this purpose, we propose a novel CSSL loss form for task t, defined as follows:

$$
\begin{aligned}
&\mathcal{L}_t^{\text{CSSL}}(\{\boldsymbol{x}^A, \boldsymbol{x}^B\}; \boldsymbol{\theta}_t, \boldsymbol{\theta}_{t-1}) \\
&= \mathcal{L}_1^{\text{SSL}^*}(\{\boldsymbol{z}_t^A, \boldsymbol{z}_t^B, \boldsymbol{z}_{t-1}^A, \boldsymbol{z}_{t-1}^B\}) \\
&+ \mathcal{L}_2^{\text{SSL}^*}(\{g(\boldsymbol{z}_t^A), \boldsymbol{z}_{t-1}^A, \boldsymbol{z}_{t-1}^B, \boldsymbol{z}_t^A, \boldsymbol{z}_t^B\}).
\end{aligned} \tag{1}
$$

Here, $g(\cdot)$ represents another MLP layer (referred to as the Predictor in the figure) introduced in (Fini et al., 2022), which has the same shape as $m_{\boldsymbol{\psi}_t}$. The SSL loss function

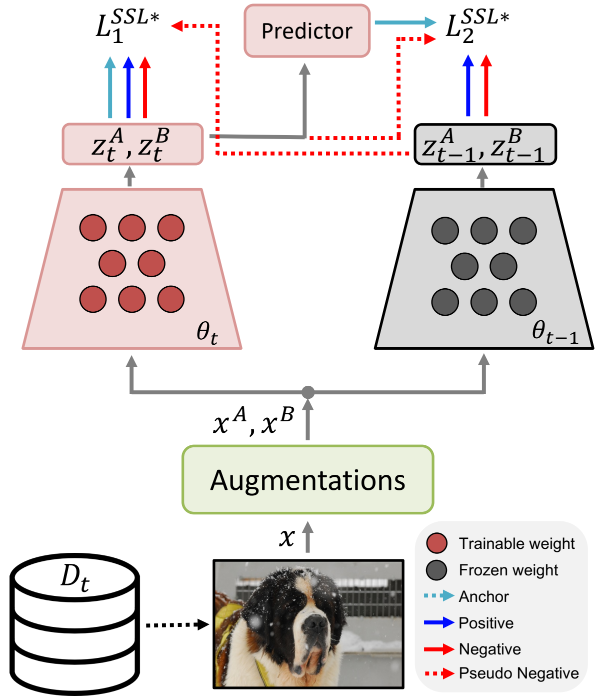


Figure 1. The overview of using pseudo-negatives in CSSL with contrastive learning. Note that red dashed arrows denote the incorporation of the proposed pseudo-negatives, which are output features from distinct models, in each loss function.

[1] For concreteness, we explicitly work with image data in this paper, but we note that our method is general and not confined to image modality.

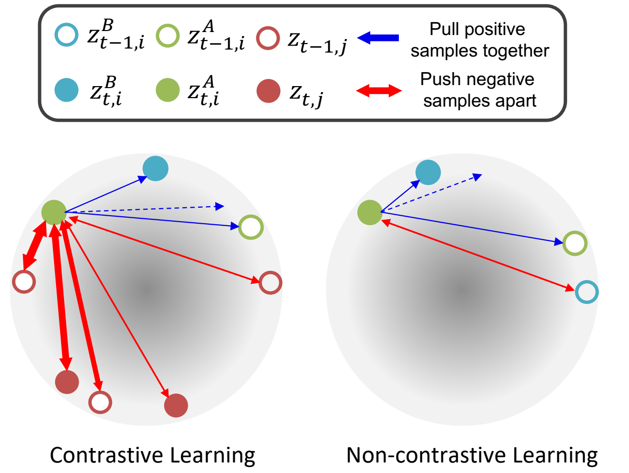


Figure 2. Graphical representation of learning with our proposed loss. The blue dashed arrow indicates the direction of the gradient update during training with the proposed loss. It moves away from the negative and pseudo-negative embeddings, which correspond to current and past models, while converging towards the positive embeddings of the current and past models.

$\mathcal{L}^{\text{SSL}^*}_{1/2}$ are newly designed general loss forms incorporating all candidate pseudo-negatives: $\boldsymbol{z}^A_{t-1}, \boldsymbol{z}^B_{t-1}$ for $\mathcal{L}^{\text{SSL}^*}_1$ and $\boldsymbol{z}^A_t, \boldsymbol{z}^B_t$ for $\mathcal{L}^{\text{SSL}^*}_2$. They resemble the two loss functions utilized in CaSSLe, and we will highlight the specific difference later. Furthermore, in order to consider two different augmentations in a symmetric fashion, we employ the following average as the final loss function for our method $\frac{1}{2}(\mathcal{L}^{\text{CSSL}}_t(\boldsymbol{x}^A, \boldsymbol{x}^B) + \mathcal{L}^{\text{CSSL}}_t(\boldsymbol{x}^B, \boldsymbol{x}^A))$.

As mentioned above, CaSSLe also uses two SSL loss functions, one responsible for plasticity and the other for stability, but they do not utilize any pseudo-negatives. Namely, the plasticity loss of CaSSLe (which corresponds to $\mathcal{L}^{\text{SSL}^*}_1$ of ours) is designed to learn new representations from the new task t and only utilizes the output features from the current model t. Moreover, the stability loss of CaSSLe (which corresponds to $\mathcal{L}^{\text{SSL}^*}_2$ of ours) aims to maintain the representations learned from the past task $t-1$ and only regularizes with the output features from the previous model $t-1$. We argue that such a stability loss term, designed to preserve representations from the previous model, might impede the plasticity loss, that aims to learn new representations from the new task but does not consider any representations from the previous model (Cha et al., 2024; Kim & Han, 2023). To address these issues, we propose utilizing the output features of both the model $t-1$ and t as the *pseudo-negatives*, that can effectively work as regularizers so that the newly learned representations may not interfere with previously learned representations.

In the upcoming section, we will specifically introduce novel function forms for $\mathcal{L}^{\text{SSL}^*}_1$ and $\mathcal{L}^{\text{SSL}^*}_2$ tailored to integrate pseudo-negatives into contrastive learning methods (e.g., SimCLR (Chen et al., 2020a), MoCo (He et al., 2020)). Additionally, we will present how the idea of using pseudo-negatives can be applied to non-contrastive learning methods (e.g., BYOL (Grill et al., 2020), VICReg (Bardes et al., 2022), BarlowTwins (Zbontar et al., 2021)) as well.

4.2. InfoNCE-based Contrastive Learning Case

Here, we propose new loss functions for CSSL using contrastive learning-based SSL methods (*e.g.*. SimCLR and MoCo). First, $\mathcal{L}^{\text{SSL}^*}_1$ in (1) is defined as:

$$\begin{aligned}&\mathcal{L}^{\text{SSL}^*}_1(\{\boldsymbol{z}^A_t, \boldsymbol{z}^B_t, \boldsymbol{z}^A_{t-1}, \boldsymbol{z}^B_{t-1}\}) \\ &= -\log \frac{\exp(\boldsymbol{z}^A_{t,i} \cdot \boldsymbol{z}^B_{t,i}/\tau)}{\sum_{\boldsymbol{z}_j \in \mathcal{N}_1(i) \cup \mathcal{PN}_1(i)} \exp(\boldsymbol{z}^A_{t,i} \cdot \boldsymbol{z}_j/\tau)},\end{aligned} \tag{2}$$

in which $\mathcal{N}_1(i) = \{\boldsymbol{z}^A_t, \boldsymbol{z}^B_t\} \backslash \{z^A_{t,i}\}$ is the set of original negatives, and $\mathcal{PN}_1(i) = \{\boldsymbol{z}^A_{t-1}, \boldsymbol{z}^B_{t-1}\} \backslash \{z^A_{t-1,i}\}$ are pseudo-negatives of $\mathcal{L}^{\text{SSL}^*}_1$. Also, $\mathcal{L}^{\text{SSL}^*}_2$ is defined as follows:

$$\begin{aligned}&\mathcal{L}^{\text{SSL}^*}_2(\{g(\boldsymbol{z}^A_t), \boldsymbol{z}^A_{t-1}, \boldsymbol{z}^B_{t-1}, \boldsymbol{z}^A_t, \boldsymbol{z}^B_t\}) \\ &= -\log \frac{\exp(g(\boldsymbol{z}^A_{t,i}) \cdot \boldsymbol{z}^A_{t-1,i}/\tau)}{\sum_{\boldsymbol{z}_j \in \mathcal{N}_2(i) \cup \mathcal{PN}_2(i)} \exp(g(\boldsymbol{z}^A_{t,i}) \cdot \boldsymbol{z}_j/\tau)},\end{aligned} \tag{3}$$

in which $\mathcal{N}_2(i) = \{\boldsymbol{z}^A_{t-1}, \boldsymbol{z}^B_{t-1}\} \backslash \{z^A_{t-1,i}\}$ is the original negatives for contrastive distillation (Tian et al., 2019; Fini et al., 2022), and $\mathcal{PN}_2(i) = \{\boldsymbol{z}^A_t, \boldsymbol{z}^B_t\} \backslash \{z^A_{t,i}\}$ is the pseudo-negatives of $\mathcal{L}^{\text{SSL}^*}_2$. Also, τ denotes a temperature parameter. In the case of SimCLR, $\mathcal{N}_1(i)$, $\mathcal{PN}_1(i)$, $\mathcal{N}_2(i)$, and $\mathcal{PN}_2(i)$ consist of negatives from the current batch. When using MoCo, two queues storing positives of each loss function ($\mathcal{L}^{\text{SSL}^*}_1$ and $\mathcal{L}^{\text{SSL}^*}_2$) from previous iterations are employed for them.

Note these two losses are quite similar in form to InfoNCE (Oord et al., 2018), have has a couple of key differences. First, in $\mathcal{L}^{\text{SSL}^*}_1$, we use pseudo-negatives in $\mathcal{PN}_1(i)$ which are obtained from the previous model $h_{\boldsymbol{\theta}_{t-1}}$. This addition of negative embeddings in $\mathcal{L}^{\text{SSL}^*}_1$ compels the embedding of $\boldsymbol{x}_i$ to be repelled not only from the negative embeddings of the current model but also from those of the previous model, hence, it fosters the acquisition of more distinctive representations. Second, in $\mathcal{L}^{\text{SSL}^*}_2$, which has the similar form of contrastive distillation, we also use pseudo-negatives in $\mathcal{PN}_2(i)$. Such modification has the impact of placing additional constraints on distillation, ensuring that the representations from the past model are maintained in a way that does not contradict the representations of the current model. Note that the denominators of the two loss functions are identical except for the $g(\cdot)$ in $\mathcal{L}^{\text{SSL}^*}_2$. Therefore, with the identical denominators, adding two losses will result in achieving a natural trade-off between plasticity and stability for learning the representation.

This intuition is depicted in the left figure of Figure 2.

Namely, both losses symmetrically consider the embeddings from current and previous models, and as shown in the hypersphere, the representation of $\boldsymbol{z}_{t,i}^A$ gets attracted to $\boldsymbol{z}_{t,i}^B$ and $\boldsymbol{z}_{t-1,i}^A$ with the constraint that it is far from $\boldsymbol{z}_{t-1,i}^B$. Thus, the new representation will be distinctive from the previous model (*plasticity*) and carry over the old knowledge (*stability*) in a way not hurt the current model.

The gradient analysis of the proposed loss function is detailed in Section A.1 of the Appendix.

4.3. Non-Contrastive Learning Case

Non-contrastive learning methods, such as Barlow, BYOL, and VICReg, do not incorporate negative samples. Therefore, the direct application of pseudo-negatives used for contrastive learning methods is not viable for these methods. However, motivated by the configuration of negatives in Equation (3), we propose a novel regularization that considers the pseudo-negatives from a new perspective. To achieve this, we propose new formulations of $\mathcal{L}_1^{\text{SSL}^*}$ and $\mathcal{L}_2^{\text{SSL}^*}$, tailored for non-contrastive learning methods, as follows:

$$\mathcal{L}_1^{\text{SSL}^*}(\{\boldsymbol{z}_t^A, \boldsymbol{z}_t^B, \boldsymbol{z}_{t-1}^A, \boldsymbol{z}_{t-1}^B\}) = \mathcal{L}^{\text{SSL}}(\{z_t^A, \boldsymbol{z}_t^B\}), \tag{4}$$

where $\mathcal{L}^{\text{SSL}}$ denotes a non-contrastive SSL loss, and

$$\begin{aligned} &\mathcal{L}_2^{\text{SSL}^*}(\{g(\boldsymbol{z}_t^A), \boldsymbol{z}_{t-1}^A, \boldsymbol{z}_{t-1}^B, \boldsymbol{z}_t^A, \boldsymbol{z}_t^B\}) \\ &= \mathcal{L}^{\text{SSL}}(\{g(\boldsymbol{z}_t^A), \boldsymbol{z}_{t-1}^A\}) - \lambda * \sum_i \sum_{\boldsymbol{z}_j \in \mathcal{PN}_2(i)} \|g(\boldsymbol{z}_{t,i}^A) - \boldsymbol{z}_j\|_2^2, \end{aligned} \tag{5}$$

where $\mathcal{PN}_2(i) = \{\boldsymbol{z}_{t-1,i}^B\}$ is the pseudo-negative, $\|\cdot\|_2^2$ represents the squared L_2 norm and λ is a hyperparameter. Additionally, $\boldsymbol{z}_{t-1}^A$ and $\boldsymbol{z}_{t-1}^B$ of Equation (4), as well as $\boldsymbol{z}_t^A$ and $\boldsymbol{z}_t^B$ of Equation (5), are not employed when using non-contrastive learning.

Note that $\mathcal{L}^{\text{SSL}}$ of Equation (5) is the CaSSLe's distillation for a non-contrstive learning method, and we introduce a new regularization to incorporate the pseudo-negatives. Specifically, $\boldsymbol{z}_{t-1,i}^B$ is assigned as the pseudo-negative of $\boldsymbol{z}_{t,i}^A$ (the anchor). This assignment stems from the configuration of negatives of Equation (3). For example, when implementing Equation (3) using SimCLR and N is the mini-batch size, $\mathcal{N}(i)$ consists of $2N-1$ negatives excluding $z_{t-1,i}^A$. Consequently, for a given anchor $\boldsymbol{z}_{t,i}^A$, an output feature $\boldsymbol{z}_{t-1,i}^B$ from the same image $\boldsymbol{x}_i$ but subjected to different augmentation is naturally considered as a negative (this holds true when using MoCo). This leads to minimizing the similarity between $g(\boldsymbol{z}_{t,i}^A)$ and $\boldsymbol{z}_{t-1,i}^B$ for the training task t. To apply this concept of negatives to CSSL using non-contrastive learning, we propose a novel regularization that maximizes the squared mean square error between $g(\boldsymbol{z}_{t,i}^A)$ and $\boldsymbol{z}_{t-1,i}^B$, ensuring their dissimilarity.

The right figure in Figure 2 illustrates representation learning with pseudo-negative in CSSL using BYOL. The CaSSLe's representation learning relies on distinct update directions from each positive (e.g., $\boldsymbol{z}_t^A$ and $\boldsymbol{z}_{t-1}^A$) to achieve enhanced plasticity and stability, without taking negatives into consideration. However, incorporating the pseudo-negative enables the model to avoid conflicts in learning representations by considering the pseudo-negative from the model $t-1$—learning representations far from the pseudo-negative from the model $t-1$. Consequently, the model acquires more distinctive representations from the previous model (enhancing plasticity) while retaining prior knowledge (ensuring stability). Note that Equation (5) can be applied in conjunction with various non-contrastive SSL methods. The implementation details for BYOL and VICReg are provided in Section A.2 of the Appendix.

We will refer to the overall framework of using pseudo-negatives for regularization in CSSL, applicable to both contrastive and non-contrastive learning as described above, as PNR (Pseudo-Negative Regularization).

5. Experiments

5.1. Experimental Details

Baselines To evaluate the proposed PNR, we set CaSSLe (Fini et al., 2022) as our primary baseline, which has shown state-of-the-art performance in CSSL. We select five SSL methods, SimCLR (Chen et al., 2020a), MoCo v2 Plus (MoCo) (Chen et al., 2020b), BarlowTwins (Barlow) (Zbontar et al., 2021), BYOL (Grill et al., 2020), and VICReg (Bardes et al., 2022), which achieve superior performance with the combination with CaSSLe.

Implementation details We implement our PNR based on the code provided by CaSSLe. We conduct experiments on four datasets: CIFAR-100 (Krizhevsky et al., 2009), ImageNet-100 (Deng et al., 2009), DomainNet (Peng et al., 2019), and ImageNet-1k (Deng et al., 2009) following the training and evaluation process outlined in (Fini et al., 2022). For CIFAR-100 and ImageNet-100, we perform class- and data-incremental learning (Class- and Data-IL) for 5 and 10 tasks (denoted as 5T and 10T), respectively. For Domain-incremental learning (Domain-IL), we use DomainNet (Peng et al., 2019), consisting of six disjoint datasets from different six source domains. Following experiments conducted by CaSSLe, we perform Domain-IL in the task order of "Real → QuickDraw → Painting → Sketch → InfoGraph → Clipart". Next, we report the average top-1 accuracy achieved by training a linear classifier separately for each domain, employing a frozen feature extractor (domain-aware evaluation). The ResNet-18 (He et al., 2016) model implemented in PyTorch is used except for ImageNet-1k where the ResNet-50 is employed. For all experiments except for ImageNet-1k, we perform each experiment using three random seeds and report the average performance across these trials. Further experimental details are available in Section C of the Appendix.

Evaluation metrics To gauge the quality of representations learned in CSSL, we conduct linear evaluation by training only the output layer on the given dataset while maintaining the encoder $h_{\boldsymbol{\theta}_t}$ as a fixed component, following (Fini et al., 2022; Cha et al., 2024). The average accuracy after learning the task t is denoted as $\mathrm{A}_t = \frac{1}{t}\sum_{i=1}^{T} a_{i,t}$, where $a_{i,j}$ stands for the linear evaluation top-1 accuracy of the encoder on the dataset of task i after the end of learning task j. Furthermore, we employ measures of stability (S) and plasticity (P), and a comprehensive explanation of these terms is provided in Section B.1 of the Appendix.

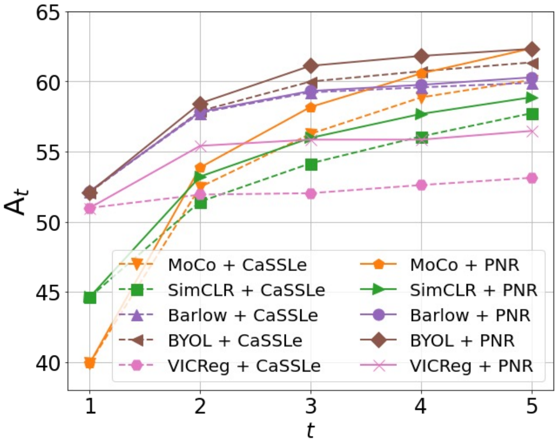


Figure 3. Experimental results of applying PNR to SSL methods. Note that "+CaSSLe" and "+PNR" indicate the results of applying CaSSLe and PNR to each SSL method, respectively.

5.2. Experiments with SSL Methods

To assess the effectiveness of PNR when applied to various SSL methods, we conduct experiments by incorporating the PNR with MoCo, SimCLR, Barlow, BTOL, and VICReg, in the Class-IL (5T) scenario using CIFAR-100. Figure 3 illustrates that both the proposed PNR and CaSSLe are successfully combined with each SSL method, demonstrating progressively enhanced the quality of representations at each task. However, PNR demonstrates more effective integration with MoCo, SimCLR, BYOL, and VICReg, surpassing the performance of CaSSLe.

Table 1. Experimental results of applying PNR to SSL methods in Class-IL (5T) with CIFAR-100. The * symbol indicates results from the CaSSLe paper, while the others are reproduced performance. The values in parentheses indicate the standard deviation and the **bolded** result represents the best performance.

A_5	MoCo	SimCLR	Barlow	BYOL	VICReg
Joint	66.90 (0.11)	63.78 (0.22)	68.99 (0.21)	69.36 (0.28)	68.01 (0.36)
FT	51.95 (0.26)	48.97 (0.74)	55.81 (0.57)	52.43 (0.62)	52.43 (0.62)
EWC*	-	53.60	56.70	56.40	-
DER*	-	50.70	55.30	54.80	-
LUMP*	-	52.30	57.80	56.40	-
Less-Forget*	-	52.50	56.40	58.60	-
+CaSSLe	60.11 (0.30)	57.73 (1.07)	60.10 (0.38)	61.36 (1.38)	53.13 (0.64)
+PNR	**62.36** (0.29)	**58.87** (0.16)	**60.28** (0.36)	**63.13** (0.42)	**56.47** (0.94)

Table 1 presents the numerical results for A_5 in the same scenario. In this table, "Joint" corresponds to the experimental results in the Joint SSL scenario (upper bound) and "FT" represents the results achieved through fine-tuning with the SSL method alone. The table reveals that PNR consistently outperforms CaSSLe, showcasing a maximum improvement of approximately 2-3%. Note that the extra performance gain achieved by PNR is particularly noteworthy, especially considering that CaSSLe's performance with MoCo, SimCLR, and BYOL is already close to that of the Joint.

5.3. Experiments with Diverse CSSL Scenarios

Class-IL Table 2 presents the experimental results of Class-IL with the CIFAR-100 and ImageNet-100 datasets. The results for Class-IL with CIFAR-100 (5T) can be found in Table 1. In comparison to the state-of-the-art method, CaSSLe, PNR demonstrates superior performance across all scenarios. Notably, the combination of PNR with MoCo, BYOL, and VICReg exhibits significantly improved performance compared to their combination with CaSSLe. For example, in the ImageNet-100 experiments, "MoCo + PNR," "BYOL + PNR," and "VICReg + PNR" achieve a substantial gain of approximately 2-6%, surpassing their combination with CaSSLe. As a result, we can confirm that the combination of PNR with each SSL method achieves state-of-the-art performance in the Class-IL scenarios.

Class-IL with ImageNet-1k Table 3 presents the experimental results for the ImageNet-1k dataset in Class-IL (5T and 10T). We only conduct experiments for MoCo and BYOL, both of which have shown superior performance in previous experiments. We train a model for each method using the same hyperparameters employed in Class-IL with the ImageNet-100 dataset. The experimental results in the table highlight the notable performance improvement of PNR in CSSL using the large-scale dataset, evident in both contrastive (MoCo) and non-contrastive (BYOL) learning methods.

Data-IL and Domain-IL Table 4 presents the experimental results in Data- and Domain-IL using the ImageNet-100 dataset. In the results of Data-IL, the combination with PNR consistently achieves state-of-the-art performance in most cases. It is noteworthy that our PNR successfully integrates

Table 2. The experimental results of Class-IL. All results are reproduced performance. The values in parentheses indicate the standard deviation and the **bolded** result represents the best performance.

A_T		Class-IL		
		CIFAR-100	ImageNet-100	
		10T	5T	10T
MoCo	Joint	66.90 (0.11)	76.67 (0.56)	
	FT	34.11 (0.90)	57.87 (0.49)	47.48 (0.42)
	+CaSSLe	53.58 (0.41)	63.49 (0.44)	52.71 (0.47)
	+PNR	**56.62** (0.31)	**67.85** (0.44)	**60.75** (0.39)
SimCLR	Joint	63.78 (0.22)	71.91 (0.57)	
	FT	39.48 (1.00)	56.11 (0.57)	46.66 (0.59)
	+CaSSLe	53.02 (0.47)	62.53 (0.11)	54.55 (0.12)
	+PNR	**53.47** (0.33)	**62.88** (0.19)	**54.79** (0.11)
Barlow	Joint	69.36 (0.21)	75.89 (0.22)	
	FT	49.46 (0.67)	60.27 (0.27)	51.83 (0.46)
	+CaSSLe	54.46 (0.24)	64.98 (0.79)	56.27 (0.63)
	+PNR	**54.69** (0.21)	**65.38** (0.81)	**56.54** (0.42)
BYOL	Joint	68.99 (0.28)	75.52 (0.17)	
	FT	46.13 (0.88)	60.77 (0.62)	51.04 (0.52)
	+CaSSLe	57.36 (0.86)	62.31 (0.09)	57.47 (0.75)
	+PNR	**58.44** (0.40)	**64.23** (0.37)	**60.11** (0.91)
VICReg	Joint	68.01 (0.36)	75.08	
	FT	46.88 (0.28)	55.58 (0.22)	46.88 (0.34)
	+CaSSLe	47.76 (0.46)	59.18 (0.36)	49.98 (0.53)
	+PNR	**49.56** (0.56)	**62.48** (0.44)	**51.82** (0.50)

Table 3. The experimental results of Class-IL with ImageNet-1k. The **bolded** result represents the best performance and we report results for a single seed.

A_T	MoCo		BYOL	
	5T	10T	5T	10T
Joint	60.62		69.46	
FT	48.33	42.55	60.76	57.15
+CaSSLe	50.57	43.38	64.78	61.93
+PNR	**56.87**	**55.88**	**66.12**	**62.56**

Table 4. The experimental results of Data and Domain-IL. All results are reproduced performance. The values in parentheses indicate the standard deviation and the **bolded** result represents the best performance.

A_T		Data-IL		Domain-IL
		ImageNet-100		DomainNet
		5T	10T	6T
MoCo	Joint	76.67 (0.56)		48.20 (0.30)
	FT	65.51 (0.72)	60.50 (0.92)	36.48 (1.01)
	+CaSSLe	66.88 (0.32)	59.72 (0.61)	38.04 (0.24)
	+PNR	**69.98** (0.30)	**67.83** (0.45)	**43.86** (0.17)
SimCLR	Joint	71.91 (0.57)		48.50 (0.21)
	FT	62.88 (0.30)	56.47 (0.11)	39.46 (0.20)
	+CaSSLe	66.05 (0.95)	61.68 (0.38)	45.96 (0.19)
	+PNR	**66.93** (0.12)	**62.04** (0.28)	**46.37** (0.13)
Barlow	Joint	75.89 (0.22)		49.50 (0.32)
	FT	66.47 (0.24)	59.48 (1.33)	41.87 (0.17)
	+CaSSLe	69.24 (0.36)	63.12 (0.28)	48.49 (0.04)
	+PNR	**70.16** (0.40)	**64.06** (0.33)	**48.90** (0.07)
BYOL	Joint	75.52 (0.17)		53.80 (0.24)
	FT	69.76 (0.45)	61.39 (0.44)	47.29 (0.08)
	+CaSSLe	**66.22** (0.13)	**63.33** (0.19)	51.52 (0.18)
	+PNR	66.08 (0.15)	63.10 (0.28)	**51.96** (0.08)
VICReg	Joint	75.08 (0.14)		52.12 (0.17)
	FT	64.02 (0.11)	57.30 (0.09)	46.11 (0.14)
	+CaSSLe	67.18 (0.15)	61.50 (0.21)	48.82 (0.12)
	+PNR	**67.68** (0.12)	**62.08** (0.20)	**48.95** (0.15)

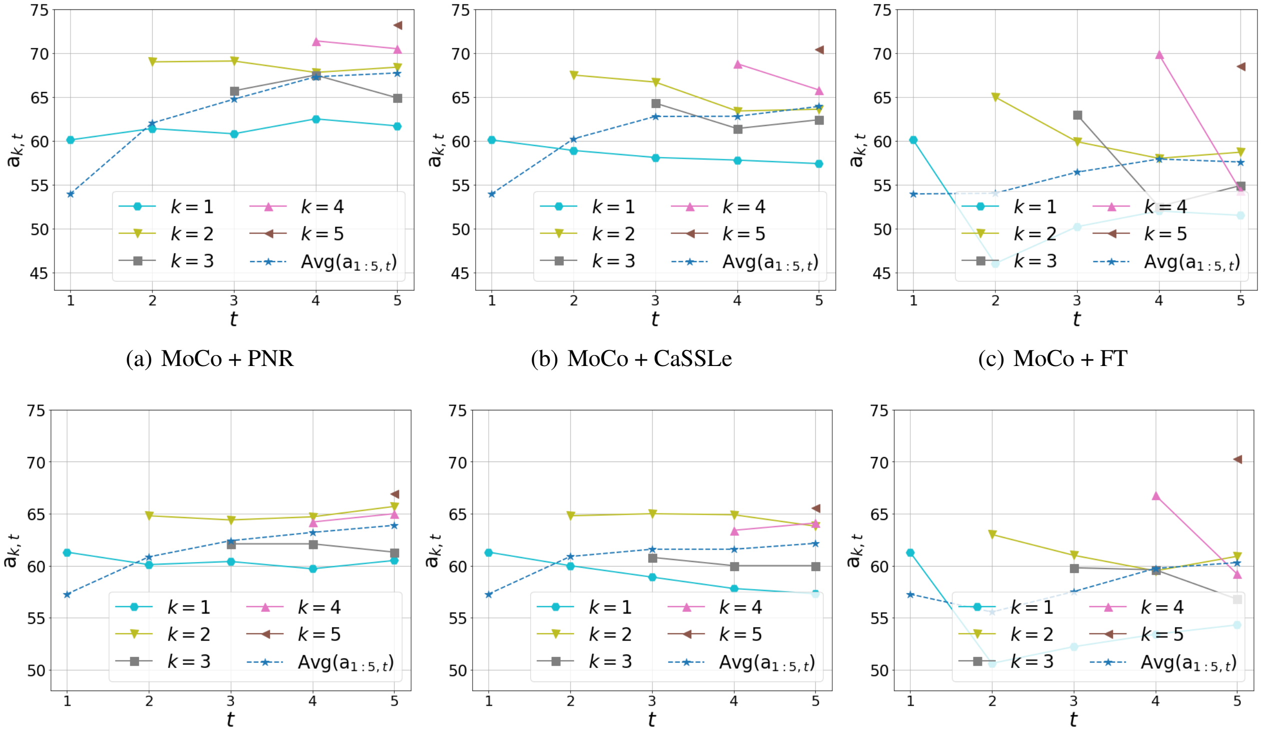
with MoCo once again, showcasing superior or nearly superior performance compared to other SSL methods at 5T and 10T. For instance, the combination "MoCo + PNR" demonstrates a 3-8% performance enhancement compared to its pairing with CaSSLe. A similar trend is observed in Domain-IL, where the combination of CaSSLe with SSL methods already demonstrates superior performance close to their respective Joint's performance. However, PNR surpasses this by achieving additional performance improvements, thus setting new standards for state-of-the-art performance. In contrast, "BYOL + PNR" performs worse than "BYOL + CaSSLe" in the Data-IL scenario. This can be attributed to the unique characteristics of Data-IL, which include shuffling and evenly distributing the ImageNet-100 dataset among tasks, leading to minimal distribution disparities between them. Additional discussion on this topic is available in Section B.2 of the Appendix.

Additionally, we conducted all experiments using CaSSLe's code and were able to reproduce the CaSSLe's reported performance on CIFAR-100. However, despite various efforts, we could only achieve performance that was 4-6% lower than the CaSSLe's reported performance in experiments using ImageNet-100 and DomainNet. More discussion regarding this matter is provided in Appendix B.3.

5.4. Experimental Analysis

Analysis on plasticity and stability Figure 4 presents the experimental results for Class-IL (5T) using the ImageNet-100 dataset, showcasing graphs of $a_{k,t}$ and Avg($a_{1:5,t}$). From Figure 4(a) showing the result of "MoCo + PNR", we observe a general upward trend in $a_{k,t}$ across all tasks. Remarkably, the performance for the initial task ($a_{k=1,t}$) remains relatively stable and even exhibits slight improvement as subsequent tasks are learned. Conversely, the results depicted in Figures 4(b) and 4(c) for "MoCo + CaSSLe" and "MoCo + FT" indicate that, while their Avg($a_{1:5,t}$) gradually increases, certain task performances experience gradual declines (e.g., $a_{k=2,t}$ of "MoCo + CaSSLe" and most k of "MoCo + FT"), showing suffering from catastrophic forgetting than "MoCo + PNR". Furthermore, the numerical assessments of plasticity (P) and stability (S) for each algorithm, as presented in the caption of Figure 4, demonstrate that "MoCo + PNR" achieves its performance improvement through superior *plasticity* and *stability* compared to other baselines.

Figures 4(d), 4(e), and 4(f) illustrate the results of our experimental analysis conducted on "PNR + BYOL," "BYOL + CaSSLe," and "BYOL + FT" in Class-IL (5T) exper-



(a) MoCo + PNR

(b) MoCo + CaSSLe

(c) MoCo + FT

(d) BYOL + PNR

(e) BYOL + CaSSLe

(f) BYOL + FT

Figure 4. The graph illustrates the values of $a_{k,t}$ for each algorithm in the Class-IL (5T) scenario using the ImageNet-100 dataset. The measured stability ($S \downarrow$) and plasticity ($P \uparrow$) for each method are as follows: (a) $(S, P) = (1.23, 3.47)$, (b) $(S, P) = (2.80, 2.52)$, (c) $(S, P) = (3.13, 2.38)$, (d) $(S, P) = (0.4, -0.07)$, (e) $(S, P) = (1.5, -0.47)$, (f) $(S, P) = (4.9, 1.6)$.

iments using the ImageNet-100 dataset. These findings align with previous results, showing a gradual increase in Avg($a_{1:5,t}$). However, notable decreases are observed in $a_{k=1,t}$ and $a_{k=3,t}$ for "BYOL + CaSSLe" across tasks, while "BYOL + FT" exhibits degraded performance in most k. In contrast, the application of our proposed PNR not only maintains $a_{k=1,t}$ effectively but also leads to a gradual increase in $a_{k=2,t}$ and $a_{k=4,t}$, indicating that our PNR outperforms "BYOL + CaSSLe" in terms of plasticity and stability. Additionally, the plasticity (P) and stability (S) measurements mentioned in the caption of Figure 4 further support these experimental findings.

In conclusion, these analyses not only highlight the effectiveness of integrating the pseudo-negative but also provide further evidence of the superior performance of our PNR. Additional analysis for other baselines can be found in Section B.4 of the Appendix.

Analysis on the impact of pseudo-negatives Table 5 presents the experimental results for different queue sizes (*i.e.*, the size of $\mathcal{PN}_1(i)$ and $\mathcal{PN}_2(i)$) in "MoCo + PNR". Note that the default queue size is 65536 for all previous experiments. We observe that performance remains relatively consistent when the queue size exceeds 16384. However, significant performance degradation is evident with reduced queue sizes (*i.e.*, reduced pseudo-negatives), particularly at 256. Based on this result, we affirm that the superior performance of "MoCo + PNR" is attributed to the utilization of a large number of pseudo-negatives.

Table 5. Experiments with different queue sizes.

Queue size	256	512	2048	16384	65536	131072
A_5	61.03	61.68	61.90	62.10	62.36	62.01

Semi-supervised learning and downstream tasks To evaluate the quality of learned representations in a more diverse way, we conduct experiments in a semi-supervised scenario. Specifically, we consider a scenario where a linear classifier is only trained using only 1% or 10% of the entire supervised ImageNet-100 dataset. We evaluate each encoder trained in the Class-IL (5T) and Data-IL (5T) using the ImageNet-100 dataset. The experimental results are presented in the upper rows of Table 6. Notably, when compared to CaSSLe, applying PNR to both MoCo and BYOL yields approximately 3-6% performance improvements in both 10% and 1%, showing new state-of-the-art performances. The lower rows of Table 6 present the results of linear evaluation for downstream tasks conducted on the same encoders. For the three datasets, we report the average accuracy of linear evaluation results on STL-10 (Coates et al., 2011), CIFAR-10, and CIFAR-100 (Krizhevsky et al., 2009) datasets. Additionally, we set Clipart within the DomainNet (Peng et al.,

Table 6. Experimental results of semi-supervised learning and downstream tasks. "10%" and "1%" denote the percent of used supervised datasets for semi-supervised learning. Also, "CIL", "DIL", "DTs" and "Cli." means "Class-IL", "Data-IL", "Downstream Tasks" and "Clipart", respectively. The **bolded** result represents the best performance

		MoCo + CaSSLe	SimCLR + CaSSLe	Barlow + CaSSLe	BYOL + CaSSLe	MoCo + PNR	BYOL + PNR
10%	CIL	56.48	55.16	55.10	54.22	**61.74**	57.36
	DIL	62.66	60.14	62.18	59.48	**65.28**	58.04
1%	CIL	39.14	40.86	41.90	36.86	**46.48**	40.82
	DIL	51.04	49.72	54.10	46.06	**54.88**	44.46
DTs	CIL	58.61	56.73	58.13	61.35	62.04	**62.53**
	DIL	59.15	56.97	59.54	61.61	**62.35**	61.95
Cli.	CIL	28.32	34.68	37.42	38.98	38.86	**41.57**
	DIL	29.74	34.17	36.13	37.04	38.33	**40.06**

2019) dataset as the downstream task and report the results of linear evaluation using it. From these experimental results, we once again demonstrate that both "MoCo + PNR" and "BYOL + PNR" achieve performance improvements compared to the CaSSLe's performance.

More detailed results and additional findings (e.g., experiments using the model trained in Class-IL (10T) and Data-IL (10T) are provided in Section B.6 of the Appendix.

Computational cost Note both PNR and CaSSLe incur almost identical computational costs, except for "MoCo + PNR," which involves an additional queue to store negatives from the $t-1$ model. However, the memory size required for this additional queue is negligible.

Ablation study Table 7 presents the results of the ablation study conducted on CIFAR-100 in the Class-IL (5T) scenario. The first row represents the performance of "MoCo + PNR" when all negatives are used. Cases 1 to 3 illustrate the results when one of pseudo-negatives, $\mathcal{PN}_1(i)$ and $\mathcal{PN}_2(i)$ is excluded or when both are omitted. The experimental results show that the absence of these negatives leads to a gradual decrease in performance, indicating that "MoCo + PNR" acquires superior representations by leveraging both the pseudo-negatives. Specifically, when $\mathcal{PN}_2(i)$ is omitted, the performance degradation is more significant compared to when the negative from $\mathcal{PN}_1(i)$ is absent. Moreover, the result of Case 4, where a queue size twice as large is used for "MoCo + CaSSLe", demonstrates that using the proposed pseudo-negatives is different from simply increasing the queue size. Note that ablation study for PNR with non-contrastive learning, such as BYOL and VICReg, can be confirmed by comparing the results of "+CaSSLe" and "+PNR" in Table 1, 2, 3, and 4.

Moreover, we present further experimental results of applying PNR to supervised contrastive learning in Section B.5 of the Appendix.

Table 7. Ablation study of "MoCo + PNR" in Class-IL (5T) using the ImageNet-100 dataset.

	$\mathcal{PN}_1(i)$	$\mathcal{PN}_2(i)$	+ CaSSLe	|queue| ×2	A_5
PNR	✓	✓	✗	✗	62.36
Case 1	✗	✓	✗	✗	61.26
Case 2	✓	✗	✗	✗	60.92
Case 3	✗	✗	✗	✗	59.97
Case 4	✗	✗	✓	✓	60.09

6. Limitation and Future Work

There are several limitations to our work. First, we focused on Continual Self-Supervised Learning (CSSL) with CNN-based architectures (e.g., ResNet). However, we believe that our proposed idea and loss function could be applied to CSSL using vision transformer-based models. Second, we only considered CSSL in the computer vision domain. Nonetheless, we believe that the concept of pseudo-negatives could be extended to CSSL in other domains, such as natural language processing. We defer these explorations to the future work.

7. Concluding Remarks

We present Pseudo-Negative Regularization (PNR), a simple yet novel method employing pseudo-negatives in Continual Self-Supervised Learning (CSSL). First, we highlight the limitations of the traditional CSSL loss formulation, which may impede the learning of superior representations when training a new task. To overcome this challenge, we propose considering the pseudo-negatives generated from both previous and current models in CSSL using contrastive learning methods. Furthermore, we expand the concept of PNR to non-contrastive learning methods by incorporating additional regularization. Through extensive experiments, we confirm that our PNR not only can be applied to self-supervised learning methods, but also achieves state-of-the-art performance with superior stability and plasticity.

Impact Statement

This paper presents a Pseudo-Negative Regularization (PNR) framework designed to advance Continual Self-Supervised Learning (CSSL). By balancing the trade-off between plasticity and stability during CSSL, our work contributes to the development of more energy-efficient AI systems, supporting both environmental sustainability and technological progress. Furthermore, our framework paves the way for more adaptive AI applications across various sectors.

Acknowledgment

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

References

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

Dosovitskiy Alexey, Philipp Fischer, Jost Tobias, Martin Riedmiller Springenberg, and Thomas Brox. Discriminative, unsupervised feature learning with exemplar convolutional, neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(9):1734–1747, 2016.

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

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2022.

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

Sungmin Cha, Hsiang Hsu, Taebaek Hwang, Flavio Calmon, and Taesup Moon. Cpr: Classifier-projection regularization for continual learning. In *International Conference on Learning Representations (ICLR)*, 2021b.

Sungmin Cha, Jihwan Kwak, Dongsub Shim, Hyunwoo Kim, Moontae Lee, Honglak Lee, and Taesup Moon. Towards more diverse evaluation of class incremental learning: Representation learning perspective. In *Third Conference on Lifelong Learning Agents (CoLLAs)*, 2024.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, pp. 1597–1607. PMLR, 2020a.

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

Haoyang Cheng, Haitao Wen, Xiaoliang Zhang, Heqian Qiu, Lanxiao Wang, and Hongliang Li. Contrastive continuity on augmentation stability rehearsal for continual self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5707–5717, 2023.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTAT)*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

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

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

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

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1422–1430, 2015.

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

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

Alex Gomez-Villa, Bartlomiej Twardowski, Kai Wang, and Joost van de Weijer. Plasticity-optimized complementary networks for unsupervised continual learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1690–1700, 2024.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:21271–21284, 2020.

Jie Gui, Tuo Chen, Qiong Cao, Zhenan Sun, Hao Luo, and Dacheng Tao. A survey of self-supervised learning from multiple perspectives: Algorithms, theory, applications and future trends. *arXiv preprint arXiv:2301.05712*, 2023.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pp. 1735–1742. IEEE, 2006.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9729–9738, 2020.

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

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

Minsoo Kang, Jaeyoo Park, and Bohyung Han. Class-incremental learning by knowledge distillation with adaptive feature consolidation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16071–16080, 2022.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:18661–18673, 2020.

Dongwan Kim and Bohyung Han. On the stability-plasticity dilemma of class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20196–20204, 2023.

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.

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

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

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

Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost Van De Weijer. Class-incremental learning: survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5513–5533, 2022.

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

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

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.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1406–1415, 2019.

Dushyant Rao, Francesco Visin, Andrei Rusu, Razvan Pascanu, Yee Whye Teh, and Raia Hadsell. Continual unsupervised representation learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2001–2010, 2017.

Chi Ian Tang, Lorena Qendro, Dimitris Spathis, Fahim Kawsar, Cecilia Mascolo, and Akhil Mathur. Kaizen: Practical self-supervised continual learning with continual fine-tuning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2841–2850, 2024.

Chenxin Tao, Honghui Wang, Xizhou Zhu, Jiahua Dong, Shiji Song, Gao Huang, and Jifeng Dai. Exploring the equivalence of siamese self-supervised learning via a unified gradient framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14431–14440, 2022.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *International Conference on Learning Representations (ICLR)*, 2019.

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

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

Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Foster: Feature boosting and compression for class-incremental learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 398–414. Springer, 2022.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

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

Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

Xiaofan Yu, Tajana Rosing, and Yunhui Guo. Evolve: Enhancing unsupervised continual learning with multiple experts. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2366–2377, 2024.

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

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 649–666. Springer, 2016.

A. Supplementary Materials for Section 3

A.1. The Gradient Analysis

Here, we give the gradient analysis of our PNR loss for contrastive learning methods. For simplicity, we assume that $g(\cdot)$ is an identity map instead of an MLP layer, hence $g(\boldsymbol{z}^A_{t,i}) = \boldsymbol{z}^A_{t,i}$. Now, we can show the gradient of $\mathcal{L}^{\text{CSSL}}_t$ with respect to $\boldsymbol{z}^A_{t,i}$ becomes

$$\frac{\partial(\frac{1}{2}\mathcal{L}^{\text{CSSL}}_t)}{\partial \boldsymbol{z}^A_{t,i}} = -\underbrace{\left(\frac{\boldsymbol{z}^B_{t,i} + \boldsymbol{z}^A_{t-1,i}}{2}\right)}_{(a)} + \underbrace{\left\{\sum_{\boldsymbol{z}_j \in \mathcal{N}_1(i) \cup \mathcal{PN}_1(i)} \boldsymbol{z}_j \cdot S^1_{t,i}(\boldsymbol{z}_j) + \sum_{\boldsymbol{z}_j \in \mathcal{N}_2(i) \cup \mathcal{PN}_2(i)} \boldsymbol{z}_j \cdot S^2_{t,i}(\boldsymbol{z}_j)\right\}}_{(b)},$$

in which $S^1_{t,i}(u) = \exp(\boldsymbol{z}^A_{t,i} \cdot u)/\sum_{\boldsymbol{z}_j \in \mathcal{N}_1(i) \cup \mathcal{PN}_1(i)} \exp(\boldsymbol{z}^A_{t,i} \cdot \boldsymbol{z}_j/\tau)$, $S^2_{t,i}(u) = \exp(\boldsymbol{z}^A_{t,i} \cdot u)/\sum_{\boldsymbol{z}_j \in \mathcal{N}_2(i) \cup \mathcal{PN}_2(i)} \exp(\boldsymbol{z}^A_{t,i} \cdot \boldsymbol{z}_j/\tau)$, and $\sum_{\boldsymbol{z}_j \in \mathcal{N}_1(i) \cup \mathcal{PN}_1(i)} S^1_{i,t}(\boldsymbol{z}_j) + \sum_{\boldsymbol{z}_j \in \mathcal{N}_2(i) \cup \mathcal{PN}_2(i)} S^2_{i,t}(\boldsymbol{z}_j) = 1$. Similarly as in the Unified Gradient of the InfoNCE loss (Tao et al., 2022), we can make the following interpretations. Namely, the negative gradient step can be decomposed into two parts, part (a) and the negative of part (b) above.

Part (a) is the average of the embedding of $h_{\boldsymbol{\theta}_t}$ for the positive sample and the embedding of $h_{\boldsymbol{\theta}_{t-1}}$ for the input sample (*i.e.,* x_i). Hence, this direction encourages the model to learn new representations while taking the *stability* from the previous model into account. On the other hand, the negative of part (b) is the repelling direction from the center of mass point among the negative sample embeddings in $\mathcal{N}_1(i) \cup \mathcal{PN}_1(i)$ and $\mathcal{N}_2(i) \cup \mathcal{PN}_2(i)$, in which each element $u \in \mathcal{N}_1(i) \cup \mathcal{PN}_1(i)$ and $u \in \mathcal{N}_2(i) \cup \mathcal{PN}_2(i)$ has the probability mass $S^1_{t,i}(u)$ and $S^2_{t,i}(u)$, respectively. Thus, this direction promotes the new representations to be more discriminative from the current and previous models' negative sample embeddings, leading to improved *plasticity*. Our gradient analysis allows us to better understand the graphical representation in the left figure of Figure 2.

A.2. PNR Implementation for Non-Contrastive Learning Methods

A.2.1. BYOL + PNR

The implementation of "BYOL + PNR" following Equation (4) and (5) is as below:

$$\mathcal{L}^{\text{SSL}^*}_1(\{\boldsymbol{z}^A_t, \boldsymbol{z}^B_t\}) = \|g_{\theta_t}(\boldsymbol{z}^A_t) - z^{\text{B}}_{\xi}\|^2_2 \tag{6}$$

$$\mathcal{L}^{\text{SSL}^*}_2(\{g(\boldsymbol{z}^A_t), \boldsymbol{z}^A_{t-1}, \boldsymbol{z}^B_{t-1}\}) = \|g(\boldsymbol{z}^A_t) - \boldsymbol{z}^A_{t-1}\|^2_2 \tag{7}$$

$$- \lambda_{\text{PNR}}\|g(\boldsymbol{z}^A_t) - \boldsymbol{z}^B_{t-1}\|^2_2, \tag{8}$$

where q_{θ_t} represents an MLP layer in BYOL, and z^{B}_{ξ} corresponds to output features of the target network obtained through momentum updates using $\boldsymbol{h}_\theta$. $\|\cdot\|^2_2$ represents the squared L_2 norm and λ_{PNR} is a hyperparameter.

A.2.2. VICReg + PNR

The implementation of "VICReg + PNR" following Equation (4) and (5) is as below:

$$\mathcal{L}^{\text{SSL}^*}_1(\{\boldsymbol{z}^A_t, \boldsymbol{z}^B_t\}) = \lambda s(\boldsymbol{z}^A_t, \boldsymbol{z}^B_t) + \mu[v(\boldsymbol{z}^A_t) + v(\boldsymbol{z}^B_t)] + \nu[c(\boldsymbol{z}^A_t) + c(\boldsymbol{z}^B_t)] \tag{9}$$

$$\tag{10}$$

$$\mathcal{L}^{\text{SSL}^*}_2(\{g(\boldsymbol{z}^A_t), \boldsymbol{z}^A_{t-1}, \boldsymbol{z}^B_{t-1}\}) = \lambda_{\text{CaSSLe}}[s(g(\boldsymbol{z}^A_t), \boldsymbol{z}^A_{t-1}) * 0.5] \tag{11}$$

$$-\lambda_{\text{PNR}}[s(g(\boldsymbol{z}^A_t), \boldsymbol{z}^B_{t-1}) * 0.5], \tag{12}$$

where Equation (9) is the original VICReg loss function and $s(\cdot,\cdot)$ denotes the mean-squared euclidean distance between each pair of vectors. λ_{CaSSLe} and λ_{PNR} are hyperparameters.

Table 8. Experimental results of different hyperparameters used for CSSL. **Bold** denotes the result of using default hyperparameters.

Hyperparameter set (for CSSL)	FT	MoCo + CaSSLe	MoCo + PNR
LR = 0.2, MBS = 128	57.34	63.56	66.50
LR = 0.4, MBS = 128	**57.52**	**62.96**	67.08
LR = 0.8, MBS = 128	56.96	55.84	67.92
LR = 0.4, MBS = 64	58.96	65.34	68.44
LR = 0.4, MBS = 256	54.20	60.50	67.80

B. Supplementary Materials for Section 4

B.1. Measures for Stability and Plasticity

To evaluate each CSSL algorithm in terms of stability and plasticity, we use measures for them, following (Fini et al., 2022; Cha et al., 2021b), as shown in below:

- Stability: $S = \frac{1}{T-1}\sum_{i=1}^{T-1} \max_{t\in\{1,\ldots,T\}}(a_{i,t} - a_{i,T})$
- Plasticity: $P = \frac{1}{T-1}\sum_{j=1}^{T-1}\frac{1}{T-j}\sum_{i=j+1}^{T}(a_{i,j} - FT_i)$

Here, FT_i signifies the linear evaluation accuracy (on the validation dataset of task i) of the model trained with a SSL algorithm for task i.

B.2. Discussion on Suboptimal Performance of CaSSLe and PNR in Data-IL

As shown in Table 4, combining BYOL with CaSSLe and PNR in Data-IL led to suboptimal performance, particularly in the 5T scenario. This can be attributed to the unique characteristics of Data-IL, briefly discussed in the CaSSLe paper. Data-IL involves shuffling and evenly distributing the ImageNet-100 dataset among tasks, resulting in minimal distribution disparities between them. During BYOL training (using Equation (6)), the target encoder (ξ) retains some information from the current task which is conceptually similar to the previous task, due to momentum updates from the training encoder ($\boldsymbol{h}_{\theta_t}$). As a result, fine-tuning solely with BYOL can produce a robust Data-IL outcome due to the substantial similarity in distribution between tasks. However, the incorporation of CaSSLe (Equation (7)) and PNR (Equation (8)) may conflict with Equation (6) in Data-IL.

On the contrary, when applied to the Domain-IL scenario using the DomainNet dataset where distinct variations in input distribution are evident for each task, "BYOL + PNR" effectively demonstrates the feasibility of augmenting negative representations as outlined in Table 4. This reinforces our conviction that the suboptimal results observed in Data-IL are solely attributable to the unique and artificial circumstances inherent to Data-IL.

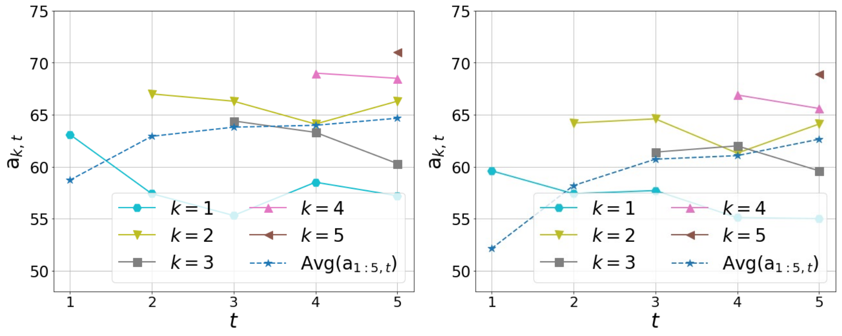
B.3. Experiments to Reproduce the CaSSLe's Reported Performance

We conduct all experiments based on the code provided by CaSSLe. For experiments on CIFAR-100, we are able to achieve results similar to those reported in the CaSSLe paper. However, despite our efforts to closely replicate the environment, including utilizing the packages and settings provided by CaSSLe, we consistently obtain results approximately 2-6% lower in experiments on ImageNet-100 and DomainNet. This issue has been raised and discussed on the issue page of CaSSLe's GitHub repository (Link 1, Link 2), where researchers have consistently reported performance discrepancies ranging from 4-6% lower than those stated in the CaSSLe paper, particularly in ImageNet-100 experiments.

Considering this discrepancy, we hypothesized that it might be related to the hyperparameter issue. Therefore, we conduct experiments by varying key hyperparameters used in linear evaluation and CSSL, such as Learning Rate (LR) and Mini-Batch Size (MBS), to address this concern. The results of these experiments are presented in Table 8 and Table 9. In Table 8, we use the CSSL model trained with default hyperparameters for CSSL, while in Table 9, we conduct experiments using the default hyperparameters for linear evaluation. Despite exploring various hyperparameter settings, we are unable to achieve the performance reported in the CaSSLe paper. However, across all considered hyperparameter configurations, our proposed "MoCo + PNR" consistently outperforms "MoCo + CaSSLe," confirming its superior performance.

Table 9. Experimental results of different hyperparameters used for linear evaluation. **Bold** denotes the result of using default hyperparameters.

Hyperparameter set (for linear eval.)	FT	MoCo + CaSSLe	MoCo + PNR
LR = 3.0, MBS = 64	57.34	64.02	68.16
LR = 3.0, MBS = 128	**57.72**	**63.88**	**68.01**
LR = 3.0, MBS = 256	57.52	62.96	67.78
LR = 3.0, MBS = 512	56.92	62.34	63.08
LR = 1.0, MBS = 256	56.42	62.00	66.52
LR = 7.0, MBS = 256	57.02	63.88	67.98



(a) Barlow + CaSSLe

(b) SimCLR + CaSSLe

Figure 5. The graph illustrates the values of $a_{k,t}$ for each algorithm in the Class-IL (5T) scenario. The measured stability and plasticity for each method are as follows: (a) $(S, P) = (2.52, 2.8)$, (b) $(S, P) = (2.22, 1.95)$.

B.4. Additional Experimental Analysis for Barlow/SimCLR + CaSSLe

Figure 5 presents an additional experimental analysis focusing on "Barlow + CaSSLe" and "SimCLR + CaSSLe" in Class-IL (5T) experiments with the ImageNet-100 dataset. Consistent with our previous observations, we once again note an improvement in their Avg($a_{1:5,t}$); however, certain tasks exhibit signs of catastrophic forgetting. Specifically, "Barlow + CaSSLe" demonstrates a decline in performance for $a_{k=1,t}$ and $a_{k=3,t}$. Similarly, $a_{k=1,t}$ and $a_{k=4,t}$ of "SimCLR + CaSSLe" follow a comparable trend.

B.5. Additional Experiments Using Supervised Contrastive Learning

We introduce Supervised Contrastive Learning (SupCon) (Khosla et al., 2020) as an additional method to investigate the effectiveness of PNR in scenarios where labels are available. To accomplish this, we modify the implementation of "SimCLR + PNR" in Equation (2) and (3) adapting it to utilize supervised labels. More specifically, it incorporates additional negatives in "SupCon + CaSSLe".

We conduct experiments with the CIFAR-100 and ImageNet-100 datasets, employing the Class-IL (5T, 10T) scenario as our experimental setting, and the results are shown in Table 10.

Table 10. Experimental results with Supervised Contrastive Learning.

A_T		CIFAR-100		ImageNet-100	
		Class-IL 5T	Class-IL 10T	Class-IL 5T	Class-IL 10T
SupCon	CaSSLe	60.38	55.38	66.26	60.48
	PNR	**60.73**	**56.16**	**66.96**	**60.95**

Based on the results in the table above, we observe that PNR can be applied to supervised contrastive learning, leading to consistent performance improvements compared to CaSSLe. We believe these experimental findings suggest the potential for the proposed PNR concept to be widely applicable across various domains using contrastive learning-based loss functions in both supervised and self-supervised manners.

B.6. Detailed Experimental Results of Downstream Tasks

Table 11. Experimental results of three downstream tasks.

Scenario		Downstream	MoCo + CaSSLe	SimCLR + CaSSLe	Barlow + CaSSLe	BYOL + CaSSLe	MoCo + PNR	BYOL + PNR
Class-IL	5T	CIFAR-100	44.52	41.99	43.59	46.71	47.8	48.8
		CIFAR-10	68.09	64.9	65.43	69.42	71.26	70.21
		STL-10	63.23	63.3	65.39	67.93	67.07	68.59
		Average	**58.61**	**56.73**	**58.13**	**61.35**	**62.04**	**62.53**
	10T	CIFAR-100	40.84	40.34	40.67	45.49	45.86	48.63
		CIFAR-10	66.01	64.62	64.42	68.02	68.65	70.34
		STL-10	60.16	59.77	62.38	65.55	65.78	67.51
		Average	**55.67**	**54.91**	**55.82**	**59.69**	**60.10**	**62.16**
Data-IL	5T	CIFAR-100	44.5	42.33	44.5	47.02	47.07	48.14
		CIFAR-10	68.47	65.03	66.92	69.14	70.01	69.03
		STL-10	64.49	63.54	67.2	68.66	69.98	68.68
		Average	**59.15**	**56.97**	**59.54**	**61.61**	**62.35**	**61.95**
	10T	CIFAR-100	42.17	41.3	43.53	45.3	46.88	48.05
		CIFAR-10	66.66	65.9	65.38	69.27	69.84	70.80
		STL-10	60.35	61.88	64.51	67.35	67.48	67.95
		Average	**56.39**	**56.36**	**57.81**	**60.64**	**61.40**	**62.27**

As emphasized in several papers (Cha et al., 2024; Chen et al., 2020a; He et al., 2020), evaluating the generalization of learned representations across diverse downstream tasks is critical. In line with this, we conduct evaluations on encoders trained with each CSSL scenario on the ImageNet-100 dataset. Following the methodology outlined in (Cha et al., 2024), we use the resized CIFAR-10/-100 datasets (resized to 96x96) (Krizhevsky et al., 2009) and the STL-10 dataset (Coates et al., 2011) as downstream tasks and performed linear evaluations on them. Table 11 showcases the exceptional performance of PNR across various downstream task datasets, consistently achieving the best overall results. Furthermore, the proposed PNR exhibits superior CSSL compared to other CaSSLe variations, particularly evident in the Data-IL scenario.

Table 12. Experimental results of Clipart in DomainNet.

Scenario		MoCo + CaSSLe	SimCLR + CaSSLe	Barlow + CaSSLe	BYOL + CaSSLe	**MoCo + PNR**	**BYOL + PNR**
Class-IL	5T	28.32	34.68	37.42	38.98	**38.86**	**41.57**
	10T	28.57	33.33	38.30	38.05	**38.19**	**38.05**
Data-IL	5T	29.74	34.17	36.13	37.04	**38.33**	**40.06**
	10T	32.81	35.53	38.45	35.91	**40.45**	**35.91**

Following the CaSSLe paper (Fini et al., 2022), we conduct additional experiments for downstream tasks. We train a model with each CSSL scenario on the ImageNet-100 dataset and conduct linear evaluation using the Clipart dataset from DomainNet as the downstream task. Table 12 presents experimental results. In the scenario of Class-IL, we observe that the models trained with "Barlow + CaSSLe" or "BYOL + CaSSLe" achieve superior performance among baselines. However, "MoCo + PNR" and "BYOL + PNR" also show competitive or state-of-the-art performance, especially in "BYOL + PNR" in Class-IL (5T). In the case of Data-IL, similar to the results obtained from previous downstream task experiments, we observe that "MoCo + PNR" and "BYOL + PNR" outperform other algorithms by a considerable margin, except for "BYOL + PNR" in Data-IL (10T).

B.7. Experimental Results Using a Different Mini-Batch Size

Table 13. Experimental results (A_5) of Class-IL (5T) with the ImageNet-100 dataset.

MoCo + FT	MoCo + CaSSLe	SimCLR + FT	SimCLR + CaSSLe	Barlow + FT	Barlow + CaSSLe	BYOL + FT	BYOL + CaSSLe	MoCo + PNR	BYOL +PNR
55.80	61.70	53.70	62.40	59.00	65.00	59.20	61.60	67.40	63.34

To compare the sensitivity of CaSSLe and PNR to mini-batch size, we train a model with a mini-batch size of 256 (default is 128) and conduct linear evaluation. The experimental results in Table 13 show slightly lower results than those mentioned in the manuscript, confirming that increasing the mini-batch size made it challenging to achieve enhanced performance. However, our PNR combination not only demonstrates similar performance across two different mini-batch sizes but also achieves superior performance in both sizes compared to CaSSLe.

C. Experimental Details

Table 14. Details on training hyperparameters used for CSSL (CIFAR-100 / ImageNet-100 / DomainNet).

CIFAR-100 / ImageNet-100 / DomainNet	MoCo (+CaSSLe)	SimCLR (+CaSSLe)	BarLow (+CaSSLe)	BYOL (+CaSSLe)	All methods + PNR
Epoch (per task)	500 / 400 / 200				
Batch size	256 / 128 / 128				
Learning rate	0.4	0.4	0.3 / 0.4 / 0.4	1.0 / 0.6 / 0.6	0.6 (for MoCo) 0.6 (for SimCLR) 0.3 / 0.4 / 0.4 (for Barlow) 1.0 / 0.6 / 0.6 (for BYOL)
Optimizer	SGD				
Weight decay	1e-4	1e-4	1e-4	1e-5	1e-5 (for BYOL) 1e-4 (for the others)
MLP Layer (dim)	2048	2048	2048	4096 / 8192 / 8192	4096 / 8192 / 8192 (for BYOL) 2048 (for the others)
Prediction layer (dim)	-	-	-	4096 / 8192 / 8192	4096 / 8192 / 8192 (for BYOL) 2048 (for the others)
Queue	65536	-	-	-	65536 (for MoCo)
Temperature (τ)	0.2	0.2	-	-	0.2 (for MoCo and SimCLR)

Table 14 presents the training details for each algorithm utilized in our Continual Self-supervised Learning (CSSL) experiments. All experiments are conducted on an NVIDIA RTX A5000 with CUDA 11.2 and we follow the experimental settings proposed in the CaSSLe's code (Fini et al., 2022). We employ LARS (You et al., 2017) to train a model during CSSL.

λ in Equation (5) Table 15 presents hyperparameter λ in Equation (5) used for experiments.

Table 15. Hyperparameter λ.

λ	CIFAR-100		ImageNet-100				DomainNet
	Class-IL		Class-IL		Data-IL		Domain-IL
	5T	10T	5T	10T	5T	10T	6T
Barlow	1	1	1	1	0.1	0.1	1
BYOL	0.5	1	1	1	0.1	0.1	1
VICReg	23	23	23	23	5	5	8

Linear evaluation Table 16 presents detailed training hyperparameters employed for linear evaluation on each dataset using encoders trained via CSSL. In the case of CaSSLe variations, we conduct the experiments while maintaining consistent linear evaluation settings with those employed in CaSSLe.

Table 16. Experimental details of linear evaluation (CIFAR-100 / ImageNet-100 / DomainNet).

CIFAR-100 / ImageNet-100 / DomainNet	MoCo (+CaSSLe)	SimCLR (+CaSSLe)	BarLow (+CaSSLe)	BYOL (+CaSSLe)	PNR
Epoch (per task)	100				
Batch size	128 / 256 / 256				
Learning rate	3.0	1.0	0.1	3.0	3.0
Scheduler	Step LR (steps = [60, 80], gamma = 0.1)				
Optimizer	SGD				
Weight decay	0				

Downstream task Table 17 outlines detailed training hyperparameters employed for linear evaluation on downstream tasks using encoders trained under various CSSL scenarios with the ImageNet-100 dataset.

Table 17. Experimental details of linear evaluation on downstream tasks.

CIFAR-10 / CIFAR-100 / STL-10 / Clipart	MoCo (+CaSSLe)	SimCLR (+CaSSLe)	BarLow (+CaSSLe)	BYOL (+CaSSLe)	PNR
Epoch (per task)	100				
Batch size	256				
Learning rate	3.0	1.0	0.1	3.0	3.0
Scheduler	Step LR (steps = [60, 80], gamma = 0.1)				
Optimizer	SGD				
Weight decay	0				