Neural Collapse Inspired Contrastive Continual Learning

Antoine Montmaur
antoine.montmaur@ensea.fr
Nicolas Larue
nicolas.larue@ensea.fr
Ngoc-Son Vu
son.vu@ensea.fr

ETIS - CY Cergy Paris Université, ENSEA, CNRS, France

Abstract

In recent advances, contrastive learning has enhanced representation quality by emphasizing transferable features across tasks, while the newly identified phenomenon of neural collapse (NC) optimizes separation capacity. Recognizing that catastrophic forgetting, the primary challenge in continual learning, results from overlapping representations between tasks, and inspired by the optimal classification ability of NC, we propose innovative strategies to minimize representational overlap. We first introduce neural continual collapse (NCC), a loss function that guides representations towards neural collapse by employing predefined hard prototypes to attract samples within a class. Additionally, we propose simplex structure distillation (SSD), a distillation technique that uses hard prototypes to strengthen knowledge consolidation. SSD improves learning stability and decreases reliance on replay buffers by gradually aligning structural distributions as tasks progress. These methods excel in challenging replay-free setups and surpass state-of-the-art (SOTA) replay-based methods.

1 Introduction

Deep neural networks have demonstrated remarkable success across various complex tasks, but their reliance on large datasets being available simultaneously for optimal performance can be a limitation. When these models face distribution shifts, their performance can degrade significantly or even fail. Continual learning (CL) offers a solution by enabling neural networks to learn from a continuous stream of evolving data. This approach allows models to adapt to changing data distributions over time while retaining previously acquired knowledge. The primary goal of CL is to minimize or prevent catastrophic forgetting (CF), a phenomenon in which the model loses prior knowledge when exposed to new data [22]. The stability-plasticity dilemma is the broader challenge of balancing the retention of previously learned knowledge with the ability to quickly acquire new tasks. CF often results from representation overlap: highly distributed representations with significant interaction can generalize well but are more prone to CF, while more localized representations with little interaction do not suffer from CF but may lack generalization ability. Semi-distributed

representations, as explored in studies like [9, 24], offer a potential compromise, though some approaches, like [24], rely on a pretrained model. An illustrative example of overlap in representations within the hypersphere can be seen in Fig. 2a.

In this context, contrastive learning has been shown to outperform supervised techniques by generating more evenly distributed representations in the representation space, thereby mitigating the overlap caused by distribution shifts. Despite this, contrastive learning methods only indirectly address catastrophic forgetting and partially tackle the root cause of overlap. Recently, neural collapse (NC) has emerged as an optimal approach for partitioning representations in output space for standard classification problems. NC uses a fixed geometric structure, a simplex of prototypes, associated with different classes. This structure optimally partitions the output space and benefits both plasticity and stability. For plasticity, it results in the most effective representations for the current task, while for stability, it provides stable anchors for representations, preventing output drift despite input drift. Building upon this insight, we integrate $\mathcal{NC}$ into contrastive continual learning by leveraging these stable prototypes. This combination aims to enhance the representation space's balance by promoting evenly distributed representations while reducing overlap.

The main contributions of this paper are as follows:

- We introduce a novel loss function called neural continual collapse (NCC), which compels representations to directly achieve neural collapse by using predefined hard prototypes as attraction anchors for all samples within a class.
- By leveraging hard prototypes, we introduce a novel distillation technique called simplex structure distillation (SSD) to enhance the consolidation of previously acquired knowledge. SSD improves stability and minimizes dependence on replay buffers by progressively aligning structural distributions as tasks are completed.
- Interestingly, our NCC and SSD strategies allow our approach to excel in challenging replay-free setup, addressing privacy concerns. Through comprehensive experiments in demanding class-incremental scenarios, we demonstrate that our replay-free method outperforms state-of-the-art (SOTA) replay-based approaches.

2 Related Work

2.1 Neural Collapse and Evenly Distributed Prototypes

During the final stage of training on a balanced dataset, the last layer's features converge towards their respective within-class averages. These within-class averages, along with the classifier vectors, then move towards the vertices of a simplex equiangular tight frame (ETF), as depicted in [23]. This configuration, proven optimal in a simplified context of last layer learning with cross-entropy loss in [12, 21], is known as neural collapse (NC). [18] also demonstrated how slight adjustments in loss functions can facilitate the emergence of such a configuration. We build upon the work of [10], who showed that NC can occur when a model is transferred to new data or classes, and [31], who applied neural collapse in a continual learning context without explicitly incorporating it into their loss function. However, our approach directly **integrates NC into the loss function**, allowing us to leverage its benefits more effectively in continual learning scenarios. We now proceed to formally present the concept of NC.

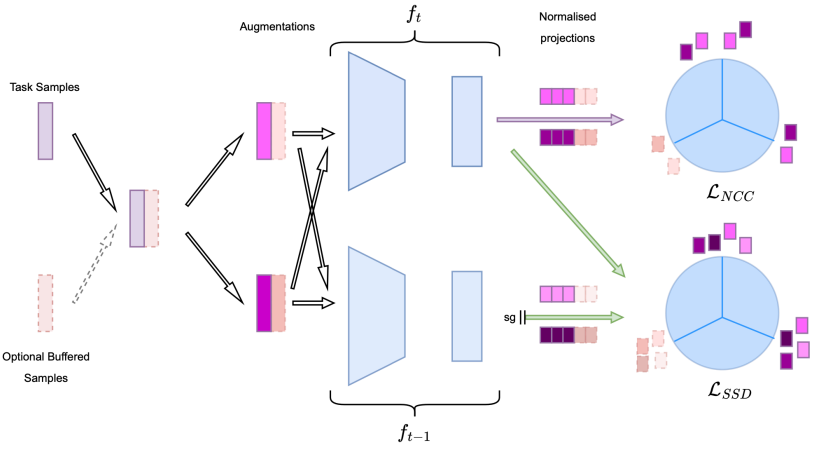

Figure 1: Overview of our framework. Augmented data are fed to both the current and previous model, then mapped to prototypes that form a simplex structure. During backpropagation, a combination of plasticity loss $\mathcal{L}_{NCC}$ and stability loss $\mathcal{L}_{SSD}$ is optimized.

Definition 1. An Equi-angular Tight Frame (ETF) is a set of L vectors $\mathbf{p}_l \in \mathbb{R}^d$, $L \in [1; d+1]$ such that $\mathbf{P} = [\mathbf{p}_1, ..., \mathbf{p}_L] \in \mathbb{R}^{d \times L}$ satisfies :

$$\mathbf{P} = \sqrt{\frac{L}{L-1}}\mathbf{U}(\mathbf{I}_L - \frac{1}{L}\mathbf{1}_L\mathbf{1}_L^\top), \tag{1}$$

where $\mathbf{U} \in \mathbb{R}^{d \times L}$ defines a rotation and $\mathbf{U}^T\mathbf{U} = \mathbf{I}_L$, $\mathbf{I}_L$ is the identity matrix and $\mathbf{1}_L$ is an all-ones vector.

All $\mathbf{p}_l$ have equal l_2 norm and pairwise angle, *i.e.*,

$$\mathbf{p}_i^\top \mathbf{p}_j = \frac{L}{L-1}\delta_{i,j} - \frac{1}{L-1}, \forall i, j \in [1, L], \tag{2}$$

where $\delta_{i,j} = 1$ if $i = j$ and 0 otherwise.

NC1: Representations from the last layer tend to their intra-class mean: the covariance matrix $\Sigma_H \to \mathbf{0}$ with $\Sigma_H := \text{Avg}_{i,k}\{(\mathbf{h}_{i,k} - \mu_k)(\mathbf{h}_{i,k} - \mu_k)^T\}$, where $\mathbf{h}_{i,k}$ is the representation of *i-th* sample of class k and μ_k is the intra-class mean of class k.

NC2: Centered intra-class means will converge to a simplex ETF, *i.e.* $\tilde{\mu}_l$, $1 < l < L$ satisfies Eq. (2) with $\tilde{\mu}_l = (\mu_l - \mu_G)/\,||\mu_l - \mu_G||$ where μ_G is the global mean.

NC3: Centered intra-class means will be colinear to corresponding classifier weights, *i.e.* $\tilde{\mu}_l$, $1 < l < L$ satisfies $\tilde{\mu}_l = \mathbf{w}_l/\,||\mathbf{w}_l||$ where $\mathbf{w}_l$ is the classifier weight associated with class l.

NC4: When NC1-NC3 holds, model prediction degenerates in similarity to class centers computation, *i.e.* $\text{argmax}_l \langle \mathbf{z}, \mathbf{w}_l \rangle = \text{argmin}_l ||\mathbf{z} - \mu_l||$ where $\mathbf{z}$ is the output of the model and $\langle , \rangle$ stands for the inner product operator.

2.2 Contrastive Learning

Contrastive learning has emerged as a leading approach in both supervised and unsupervised learning domains. Recent work [15, 29] demonstrates that contrastive methods achieve per-

formances on par with models trained using cross-entropy loss, as highlighted in [12]. These methods often involve augmentations of each data sample (positive pair), training models to generate similar representations for these pairs by ensuring invariance to the augmentations. Given a feature extractor f and a linear projector g, the model's output for a data point $\mathbf{x}_i$ is represented by $\mathbf{z}_i = h(\mathbf{x}_i) = g \circ f(\mathbf{x}_i)$. Generic representations arise through the alignment of positive pairs, which are pulled together in clusters, and the uniformity created by pushing apart negative pairs, thereby separating the clusters, as initially formulated in [15].

2.3 Continual Learning

Continual learning is explored in various scenarios typically grouped into three categories: task-incremental, domain-incremental, and class-incremental [28]. In our work, we focus on task-incremental learning (T-IL) and class-incremental learning (C-IL). In these scenarios, the learner is trained on a series of tasks indexed by $t \in 1, 2, \ldots, T$. Each task involves a specific class set C_t, which are assumed to be disjoint across tasks. This disjoint assumption can be formally stated as: if $t \neq t'$, then $C_t \cap C_{t'} = \emptyset$ for T-IL and C-IL.

During each task, n_t input-label pairs are independently drawn from some task-specific distribution, *i.e.*, $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n_t} \sim D_t$. Here, $\mathbf{x}$ denotes the input image, and $\mathbf{y}_i \in C_t$ denotes the class label belonging to the task-specific class set. For T-IL, the learned models can access each task label t during test phase; the goal is to find a model $f_\theta(\mathbf{x}, t)$ parameterized by θ such that a combination of each task objectives is minimized for some loss function $\ell(\cdot, \cdot)$. For C-IL, the only difference lies in the absence of t in $f_\theta(\mathbf{x})$ as the model cannot access task identities.

Many strategies have been proposed to address catastrophic forgetting [2, 5, 7, 8, 9, 11, 14, 16, 20, 25, 26, 27]. These methods generally fall into five categories: regularization, replay, optimization, representation, and architecture. We concentrate on optimization-based methods [5, 20] by specifically crafting an optimization program tailored for a contrastive learning context. A generic form of any continual optimization function is the combination of a function dedicated to learning new knowledge for the current task and another function for regularizing and stabilizing knowledge from previous tasks, resulting in:

$$\mathcal{L}_{continual} = \mathcal{L}_{plasticity} + \mathcal{L}_{stability} \tag{3}$$

More specifically, continual contrastive learning has recently emerged as an optimization approach, introduced by Cha *et al.* in [3] for supervised learning and by Fini *et al.* in [8] for unsupervised learning. In these studies, the plasticity loss $\mathcal{L}_{plasticity}$ is implemented as either supervised contrastive learning (scl) [15] or SimCLR [6]. Other works [1, 19] employ prototypes to develop a distillation scheme for stability loss $\mathcal{L}_{stability}$, as defined in Eq. (3), using these prototypes as anchors to stabilize representations during training. However, these prototypes are learned, making them susceptible to drift between tasks. Additionally, some methods, such as [19], still rely on replay buffers to mitigate forgetting.

3 Methodology

While NC has been proven advantageous for representation learning, most approaches [23, 31] aim to achieve it by substituting the model's last layer with a static ETF classifier, also known as a cosine classifier. In practice, nearest class mean based predicted label $\hat{y}_i$ for sample i is obtained via: $\hat{y} = \text{argmax}_l(\tilde{\mathbf{z}}_i^T \tilde{\boldsymbol{\mu}}_l)$, where $\tilde{\mathbf{z}}_i^T$ represents the normalized representation

$\frac{\mathbf{z_i^T}}{\|\mathbf{z_i^T}\|}$, and similarly, $\tilde{\mu}_l = \frac{\mu_l}{\|\mu_l\|}$, simplifying predictions but scarcely improving training as losses rely on class centroids rather than representations.

Unlike these works, we propose a more effective approach that directly aligns representations with their specific geometry by employing evenly distributed prototypes. First, in Section 3.1, we explain how to generate evenly distributed prototypes, then detail the novel stability ($\mathcal{L}_{NCC}$) and plasticity ($\mathcal{L}_{SSD}$) losses in Sec. 3.2 and Sec. 3.4.

3.1 Evenly Distributed Prototypes

Both new NC-inspired losses contract upon evenly distributed prototypes forming a simplex, denoted as $[\mathbf{p}_1, \ldots, \mathbf{p}_L] \in \mathbb{R}^{d \times L}$, consisting of evenly distributed points. In mathematical geometry and optimization, the simplex is a foundational construct, defined as the generalization of a triangle into higher dimensions, a structure visible in dimension 2 in Fig. 1 and Fig. 2. In an n-dimensional space, a simplex is the convex hull of $(n+1)$ affinely independent points. Creating a simplex with evenly distributed points requires careful arrangement to ensure equidistance and symmetrical positioning among vertices. This configuration aims to form an ETF, creating a geometric structure where each point contributes equitably to the simplex's shape and properties. We use the algorithm developed in [17] for instantiating points. We provide this algorithm in appendix A of supplementary material.

3.2 Neural Collapse guided Loss

We use simplex points as hard prototypes for each class and aim to map all representations from a specific class to the same prototype. It is common practice to guide a model towards a desired output by formulating a loss function that encourages it. Thus, to facilitate the emergence of the desired ETF geometry, we introduce prototypes forming the ETF in our loss function. Building on this intuition, we propose a new loss that extends the established supervised contrastive loss [15], incorporating the set of $\mathbf{p}_l$ as defined in Sec. 3.1 to achieve the structure described in Sec. 2.1. Our loss takes the following form:

$$\mathcal{L}_{NCC} = \sum_{i=1}^{2N} \ell_{NCC}(\mathbf{x}_i) = \sum_{i=1}^{2N} \frac{-1}{|P(i)|} \left[\sum_{p \in P(i)} \log \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_p/\tau)}{\sum_{\substack{j=1 \\ j \neq i}}^{2N} \exp(\mathbf{z}_i \cdot \mathbf{z}_j/\tau)} + \log \frac{\exp(\mathbf{z}_i \cdot \mathbf{p}_{\mathbf{y}_i}/\tau)}{\sum_{\substack{j=1 \\ j \neq i}}^{L} \exp(\mathbf{z}_i \cdot \mathbf{p}_{\mathbf{y}_j}/\tau)} \right] \tag{4}$$

where $P(i) \equiv \{j \in [1,2N], j \neq i : \mathbf{y}_j = \mathbf{y}_i\}$ is the set of all positive pairs for sample i in the multiviewed batch, $|P(i)|$ is its cardinality and $\mathbf{p}_{\mathbf{y}_i}$ represents the ETF vertex mapped to $\mathbf{y}_i$, the label of sample i. The goal is to cluster representations from the same class around a shared prototype. The second term $\ell_{proto}(\mathbf{x}_i)$ in Eq. (4) functions like a scaled softmax, considering all prototypes, achieving a similar effect on the positioning of samples relative to prototypes as is done for the alignment of samples with one another by $\ell_{scl}(\mathbf{x}_i)$ in [15]. This method strengthens the generated features by directing them towards their optimal average position. As shown in Fig. 2b, representations gain from fixed prototypical areas, which bring positive pairs closer by providing a shared anchor around their prototype. To our knowledge, this loss is the first of its kind implemented to achieve NC within contrastive continual learning.

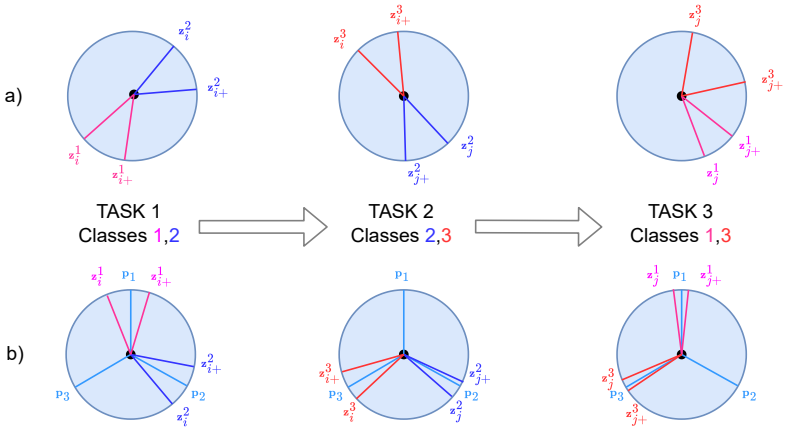

Figure 2: In classical contrastive continual learning, representations may overlap across tasks, as shown in a). With optimal equidistant prototypes, b) minimizes overlap in representations through consistent assignment.

3.3 Gradient Analysis

Here, we will make some remarks on our $\mathcal{L}_{NCC}$ based on its derivation with respect to a sample $\mathbf{z}_i$. For clarity, let's first recall that Eq. (4) represents a weighted sum over all positive samples $\mathbf{z}_i$ of the loss $\ell_{NCC}(\mathbf{z}_i)$, as defined in Eq. (5). Without loss of generality, we will consider the case where only one positive pair is formed with the sample $\mathbf{z}_p$. Furthermore, we will simplify the sum indices, irrespective of iteration over $2N$ samples or L prototypes.

$$\ell_{NCC}(\mathbf{z}_i) = \log \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_p/\tau)}{\sum_{j\neq i} \exp(\mathbf{z}_i \cdot \mathbf{z}_j/\tau)} + \log \frac{\exp(\mathbf{z}_i \cdot \mathbf{p}_{y_i}/\tau)}{\sum_{j\neq i} \exp(\mathbf{z}_i \cdot \mathbf{p}_{y_j}/\tau)} \tag{5}$$

where y_i is sample i's label. For convenience, we defer the proof of **Proposition 1.** to appendix B of supplementary material.

Proposition 1. The gradient of $\ell_{NCC}(\mathbf{z}_i)$ shares similar structure for sample-to-sample and sample-to-prototype parts as in Eq. (6).

$$\nabla_{\mathbf{z}_i} \ell_{NCC}(\mathbf{z}_i) = \frac{1}{\tau}[P_i(\mathbf{z}_p) - N_i(\mathbf{z}_j) + P_i(\mathbf{p}_{y_i}) - N_i(\mathbf{p}_{y_j})] \tag{6}$$

where P and N represent the positive and negative contributions to SGD, respectively.

The roles of the terms P_i and N_i can be summarized as follows: Both P_i terms act as pulling forces; however, $P_i(\mathbf{p}_{y_i})$ remains static. Additionally, it is important to note that both N_i terms represent weighted means of pushing forces, where the weighting coefficients are either samples or prototypes themselves. Here again, $N_i(\mathbf{p}_{y_j})$ serves as a static pushing term with respect to each $\mathbf{z}_i$. In conclusion, prototypes exhibit an accumulating and stabilizing effect on $\mathcal{L}_{scl}$.

3.4 Simplex Structure-based Distillation

By analogy to the scheme described in Eq. (3), we introduce a stability-preserving approach enabled through simplex structure distillation (SSD). Given our current contrastive model

$h_t = g_t \circ f_t$ from Sec. 2.2 and its predecessor $h_{t-1} = g_{t-1} \circ f_{t-1}$, we represent their outputs as $\mathbf{z}_i^t$ and $\mathbf{z}_i^{t-1}$, denoting $h_t(\mathbf{x}_i)$ and $h_{t-1}(\mathbf{x}_i)$ respectively.

Our method leverages the Kullback-Leibler (KL) divergence to compare distributions of feature relationships to prototypes. This approach aligns with existing work such as [1], which used KL divergence for knowledge distillation to map learned prototypes to features. However, our formulation offers a new perspective, detailed as follows:

$$\mathcal{L}_{SSD} = \sum_{i=1}^{N} \mathcal{D}_{KL}(P(\mathbf{Z}^{t-1}; \mathbf{p}_i) || P(\mathbf{Z}^t; \mathbf{p}_i)), \tag{7}$$

where $P(\mathbf{Z}^{t-1}; \mathbf{p}_i)$ stands for softmax distribution of cosine similarities between samples in tasks and a given prototype for both current and previous representations. Unlike conventional knowledge distillation methods, our approach preserves the overall structure of samples across different tasks without enforcing alignment with previous distributions. This is achieved by optimally placing hard prototypes that remain fixed as new continual tasks emerge, allowing flexibility in representations for new classes. In Fig. 2b, average position of representations per class is maintained through prototypes, while in Fig. 2a, representations may diverge from their original positions. Ensuring appropriate distribution of representations across tasks is essential since only relevant prototypes, selected based on their label, are considered by $\mathcal{L}_{SSD}$ at each task step. In contrast, Co^{2}L [3] utilizes instance-wise distillation, which directly aligns the model's output with those of previous iterations. Meanwhile, in PRD [1], both the current encoder and prototypes are simultaneously trained, potentially leading to representation drift and an increased risk of forgetting.

Our final objective can be expressed as follows:

$$\mathcal{L}_{final} = \mathcal{L}_{plasticity} + \mathcal{L}_{stability} = \mathcal{L}_{NCC} + \alpha \mathcal{L}_{SSD} \tag{8}$$

where α is a hyperparameter we simply set to 1 for most experiments. For completeness, we will discuss effects of this factor in Sec. 4.

4 Experiments

4.1 Implementation details

We utilize a ResNet-18 backbone [13], and following [3, 8], we remove the last layer of the backbone. Building on work from [6, 32, 33] and others, we add a multi-layer perceptron head with linear layers and ReLU activation functions after the backbone, in line with standard architectures. We train the backbone for 500 epochs per task using SGD, then separately train a linear classifier using samples from the last task and buffered samples. We assess performance on CIFAR-10 and Tiny-ImageNet across 5 tasks with 2 classes each and 10 tasks with 20 classes each, respectively. Following [3], we test buffer sizes of 200 and 500, as well as a replay-free setting with a buffer size of zero.

4.2 Metrics

To validate our hypothesis, we use the commonly employed metric of *average accuracy* as defined in [4, 30] to measure overall model performance across tasks. Let $a_{k,j} \in [0,1]$ represent the classification accuracy evaluated on the test set of the j-th task after incremental

Buffer	Method	Seq-CIFAR-10		Seq-Tiny-ImageNet	
		C-IL	T-IL	C-IL	T-IL
200	GEM [20]	25.54±0.76	90.44±0.94	-	-
	A-GEM [6]	20.04±0.34	83.88±1.49	8.07±0.08	22.77±0.03
	iCaRL [25]	49.02±3.20	88.99±2.13	7.53±0.79	28.19±1.47
	DER [2]	61.93±1.79	91.40±0.92	11.87±0.78	40.22±0.67
	DER++ [2]	64.88±1.17	91.92±0.60	10.96±1.17	40.87±1.16
	Co^{2}L [3]	65.57±1.37	93.43±0.78	13.88±0.40	42.37±0.74
	Ours	**68.97**±1.28	**95.04**±0.73	**15.32**±0.52	**43.57**±0.41
500	GEM [20]	26.20±1.26	93.81±0.27	-	-
	A-GEM [6]	22.67±0.57	89.48±1.45	8.06±0.04	25.33±0.49
	iCaRL [25]	47.55±3.95	88.22±2.62	9.38±1.53	31.55±3.27
	DER [2]	70.51±1.65	93.40±0.39	17.75±1.14	51.78±0.88
	DER++ [2]	72.70±1.36	93.88±0.50	19.38±1.41	51.91±0.68
	Co^{2}L [3]	74.26±0.77	95.90±0.26	20.12±0.42	53.04±0.69
	Ours	**75.83**±0.79	**97.02**±0.59	**21.51**±0.35	**54.96**±0.43

Table 1: Accuracies on Seq-CIFAR-10 and Seq-Tiny-ImageNet datasets. Results are reported as mean ± standard deviation across five trials.

	Buffer size		
Method	0	200	500
Co^{2}L with $\mathcal{L}_{SCL}^{asym}$ & $\mathcal{L}_{IRD}$ [3]	53.57±1.03	65.57±1.37	74.26±0.77
$\mathcal{L}_{NCC}$ & $\mathcal{L}_{IRD}$	61.81±1.22	67.72±1.17	75.14±0.67
$\mathcal{L}_{NCC}$ & $\mathcal{L}_{SSD}$	**65.96**±1.17	**68.97**±1.28	**75.83**±0.79

Table 2: Accuracies on CIFAR-10 dataset across 5 tasks in a C-IL scenario, demonstrating that our replay-free method can surpass existing replay-based approaches [3] with a buffer size of 200. IRD stands for Instance-wise Relation Distillation [3].

learning of the k-th task ($j \leq k$). The output space for computing $a_{k,j}$ includes either the classes in Y_j or $\cup_{i=1}^{k} Y_i$, depending on whether multi-head (e.g., TIL) or single-head (e.g., CIL) evaluation is used [4]. The *AA* metric at the k-th task is then calculated as: $AA_k = \frac{1}{k}\sum_{j=1}^{k} a_{k,j}$, representing the current overall performance.

4.3 Results

As can be seen in Tab. 1, in the challenging class-incremental learning (C-IL) scenario, our method outperforming all baselines by 3% on the CIFAR-10 dataset and 1.5% on Tiny-ImageNet. Additionally, for the simpler task-incremental learning (T-IL) scenario, our method consistently surpasses all baselines by 2% and 1% respectively. These findings emphasize our approach's adaptability across various scenarios, enabling it to acquire new knowledge while effectively retaining previously learned information.

To thoroughly compare with a state-of-the-art contrastive continual learning method, we evaluate our components against those in [3] as detailed in **??**. Our results show that $\mathcal{L}NCC$ consistently outperforms $\mathcal{L}SCL$ from [3], even without a replay buffer, where it improves accuracy by 8%. This gain is due to the simplex structure's independence from replay schemes, with memory encoded in fixed prototype positions. Furthermore, our stability loss's distributional nature enhances adaptability by avoiding the rigid replication of past model behavior.

α	0.25	0.5	1	2	5
$\mathcal{L}_{NCC}$ & $\mathcal{L}_{IRD}$	62.33±1.43	65.26±1.12	67.72±1.17	64.23±1.05	59.15±1.97
$\mathcal{L}_{NCC}$ & $\mathcal{L}_{SSD}$	65.96±1.22	68.07±1.09	**68.97**±1.28	66.39±0.79	61.22±0.92

Table 3: Ablation studies on hyperparameter α with both $\mathcal{L}_{IRD}$ and $\mathcal{L}_{SSD}$. Experiments are conducted with a buffer of 200 samples.

Description	Buffer size	Stability	Accuracy (%)
$\mathcal{L}_{NCC}$	0		52.41±1.13
$\mathcal{L}_{NCC}$ & $\mathcal{L}_{SSD}$	0	✔	65.96±1.17
$\mathcal{L}_{NCC}$	200		55.27±0.79
$\mathcal{L}_{NCC}$ & $\mathcal{L}_{SSD}$	200	✔	**68.97**±1.28

Table 4: Ablation study showing the necessity of stability loss.

4.4 Ablation Studies

First, we examine the impact of the hyperparameter α on our performance. Initially, following [8], we did not modify its value. However, we later conducted experiments to analyze the influence of this parameter. We selected five values of α and applied our standard training protocol with a buffer size of 200 samples.

The results, summarized in Tab. 3, indicate that $\alpha = 1$ is optimal for all regularization losses, confirming our primary setting. Notably, $\mathcal{L}_{SSD}$ consistently outperforms $\mathcal{L}_{IRD}$ across all α values, as detailed in Tab. 4. Additionally, sensitivity to α appears to be lower with $\mathcal{L}_{SSD}$, as reflected by the generally smaller decrease in performances.

To complete our work, we present an ablation study highlighting the critical role of $\mathcal{L}_{SSD}$ in Tab. 4. The study reveals a significant performance drop when stability loss is absent, regardless of buffer size. This decline occurs because the model lacks insight into how its previous version processed samples, a perspective not conveyed solely through replay buffers. Moreover, even if a drop in performances is observed without a buffer, results remain high when using $\mathcal{L}_{SSD}$.

4.5 Prototypes Analysis

To validate our hypothesis that prototypes facilitate both learning new representations and preserving their distribution, we analyze the relationship between prototypes and representations in Fig. 3. The histograms in Fig. 3 were generated from a standard C-IL scenario on the CIFAR-10 dataset. In the absence of replay samples, cosine similarities between all class samples and their respective prototypes were calculated using the representations from the most recent model.

Despite the distribution shifts, the final model accurately maps old samples to their prototypes, showing the robustness of representations learned with $\mathcal{L}_{NCC}$. Class centroids (dotted lines) align closely with hard prototypes, suggesting effective representation alignment. Sample distributions around prototypes follow a Gaussian pattern centered on class centroids, indicating stable representations across tasks due to $\mathcal{L}_{NCC}$. Lastly, a direct correlation between proximity to prototypes and per-class accuracy is evident, especially in the "cat" class, where greater distance from the prototype results in lower accuracy.

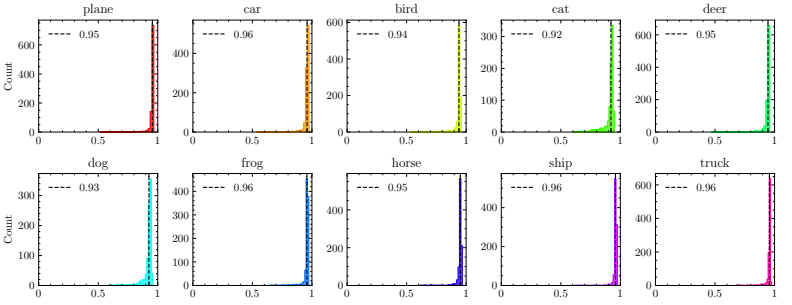

Figure 3: Histogram showing cosine similarity for each class's features compared to its prototype. The x-axis represents cosine similarity, while the y-axis indicates the bin count. The dotted lines mark the computed class centroids, and the per-class accuracy is shown alongside. A cosine similarity of 1 signifies a perfect collapse to the class prototype.

5 Conclusion

We demonstrated the potential of inducing NC to enhance both stability and plasticity in continual learning settings, addressing the historical trade-off between them. We achieved this by implementing two effective loss functions that leverage predefined prototypes and testing them across various continual learning scenarios. The first loss, $\mathcal{L}_{NCC}$, steers representations towards neural collapse by using predefined hard prototypes to draw in samples within a class. The second loss, $\mathcal{L}_{SSD}$, improves learning stability and reduces reliance on replay buffers by gradually aligning structural distributions as tasks evolve. In future work, we plan to delve deeper into how clusters form around prototypes. This exploration could provide insights into our model's confidence level, allowing us to identify samples that are more prone to being forgotten and adjust our loss functions to account for this new parameter.

6 Acknowledgement

This work was partially supported by funding from the Agence Innovation Défense (AID) and utilized HPC resources from GENCI- IDRIS (Grant AD011013944R1).

References

[1] Nader Asadi, MohammadReza Davari, Sudhir Mudur, Rahaf Aljundi, and Eugene Belilovsky. Prototype-sample relation distillation: Towards replay-free continual learning. In *ICML*, 2023.

[2] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *NeuRIPS*, 2020.

[3] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co^{2}l: Contrastive continual learning. In *CVPR*, 2021.

[4] Arslan Chaudhry, Puneet K. Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *ECCV*, 2018.

[5] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *ICLR*, 2019.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[7] Aristotelis Chrysakis and Marie-Francine Moens. Online bias correction for task-free continual learning. In *ICLR*, 2023.

[8] Enrico Fini, Victor G. Turrisi da Costa, Xavier Alameda-Pineda, Elisa Ricci, Karteek Alahari, and Julien Mairal. Self-supervised models are continual learners. In *CVPR*, 2022.

[9] Robert French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3:128–135, 05 1999. doi: 10.1016/S1364-6613(99)01294-2.

[10] Tomer Galanti, András György, and Marcus Hutter. On the role of neural collapse in transfer learning. In *ICLR*, 2022.

[11] Yasir Ghunaim, Adel Bibi, Kumail Alhamoud, Motasem Alfarra, Hasan Abed Al Kader Hammoud, Ameya Prabhu, Philip H.S. Torr, and Bernard Ghanem. Real-time evaluation in online continual learning: A new hope. In *CVPR*, 2023.

[12] Florian Graf, Christoph D. Hofer, Marc Niethammer, and Roland Kwitt. Dissecting supervised contrastive learning. In *ICML*, 2023.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2015.

[14] Zhiyuan Hu, Yunsheng Li, Jiancheng Lyu, Dashan Gao, and Nuno Vasconcelos. Dense network expansion for class incremental learning. In *CVPR*, 2023.

[15] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeuRIPS*, 2020.

[16] Dongwan Kim and Bohyung Han. On the stability-plasticity dilemma of class-incremental learning. In *CVPR*, 2023.

[17] Kenneth Lange and Tong Tong Wu. An mm algorithm for multicategory vertex discriminant analysis. *Journal of Computational and Graphical Statistics*, 17(3):527–544, 2008. doi: 10.1198/106186008X340940.

[18] Nicolas Larue, Ngoc-Son Vu, Vitomir Struc, Peter Peer, and Vassilis Christophides. Seeable: Soft discrepancies and bounded contrastive learning for exposing deepfakes. In *ICCV*, 2023.

[19] Huiwei Lin, Baoquan Zhang, Shanshan Feng, Xutao Li, and Yunming Ye. Pcr: Proxy-based contrastive replay for online class-incremental continual learning. In *CVPR*, 2023.

[20] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *NeuRIPS*, 2022.

[21] Jianfeng Lu and Stefan Steinerberger. Neural collapse with cross-entropy loss. *CoRR*, abs/2012.08465, 2020.

[22] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation - Advances in Research and Theory*, 24(C):109–165, January 1989. ISSN 0079-7421. doi: 10.1016/S0079-7421(08)60536-8.

[23] Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, September 2020. ISSN 1091-6490.

[24] Julien Pourcel, Ngoc-Son Vu, and Robert M. French. Online task-free continual learning with dynamic sparse distributed memory. In *ECCV*, 2022.

[25] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017.

[26] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *ICLR*, 2019.

[27] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks, 2022.

[28] Gido M. van de Ven and Andreas S. Tolias. Three scenarios for continual learning, 2019.

[29] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019.

[30] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application, 2023.

[31] Yibo Yang, Haobo Yuan, Xiangtai Li, Zhouchen Lin, Philip Torr, and Dacheng Tao. Neural collapse inspired feature-classifier alignment for few-shot class incremental learning. In *ICLR*, 2023.

[32] Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. Decoupled contrastive learning. In *ECCV*, 2022.

[33] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *ICML*, 2021.