Adaptive Class Aware Memory Selection and Contrastive Representation Learning for Robust Online Continual Learning in both Balanced and Imbalanced Data Environments

Rui Yang, Matthieu Grard, Emmanuel Dellandréa, Liming Chen

Adaptive Class Aware Memory Selection and Contrastive Representation Learning for Robust Online Continual Learning in both Balanced and Imbalanced Data Environments

Rui Yang[1], Matthieu Grard[2], Emmanuel Dellandrea[1], Liming Chen[1]
[1]Ecole Centrale de Lyon, CNRS, Universite Claude Bernard Lyon 1
INSA Lyon, Université Lumière Lyon2, LIRIS, UMR5205, 69130 Ecully, France
[2]Siléane, 17 rue Descartes, 42000 Saint-Etienne, France

***Abstract*—Online Continual Learning (OCL) is a framework where models learn continuously from a stream of data without revisiting previously seen data. This is crucial for many real-life applications, *e.g.*, autonomous driving, healthcare monitoring, and robotics, where data evolves over time. However, current state-of-the-art continuous learning methods struggle with dynamic and unbalanced data, often failing to adapt and leading to severe performance degradation. In this paper, we introduce Memory Selection with Contrastive Learning (MSCL), an advanced approach to Continual Learning (CL) designed to tackle these challenges. MSCL integrates Feature-Distance Based Sample Selection (FDBS) for effective memory adaptation, emphasizing inter-class similarities and intra-class diversity, with a novel contrastive learning loss (SCL) for evolving data representation consolidation. Our extensive evaluations on datasets including CIFAR-100, Mini-ImageNet, PACS, and DomainNet demonstrate that MSCL not only surpasses existing memory-based CL methods on data balanced scenarios, but also excels particularly in imbalanced scenarios, thereby establishing a novel state of the art in both balanced and imbalanced learning contexts. Additionally, we carefully conduct ablation studies to highlight the contribution of each component, *i.e.*, FDBS and SCL, and analyze the impact of the key hyperparameter, *i.e.*, memory size, on the performance of the proposed MSCL method.**

***Index Terms*—Continual Learning, Transfer Learning, Memory Selection**

I. Introduction

Continual Learning (CL) involves a model learning from a continuous stream of data over time without access to previously encountered data, posing the challenge of *catastrophic forgetting*—the loss of previously acquired knowledge when new information is learned.

Current CL methods fall into three main categories: Regularization-based approaches [1], [2], Parameter Isolation approaches [3], [4], and Rehearsal-based approaches [5]–[8]. Various CL paradigms have also been explored [9], such as Task-Incremental Learning (TIL), Domain-Incremental Learning (DIL), and Class-Incremental Learning (CIL). Early CL methods like [1], [10] primarily used the TIL paradigm, assuming access to task boundaries during both training and inference, which is often unrealistic. Consequently, recent research has shifted towards CIL [11]–[13], where models learn from sequential, mutually exclusive class tasks and infer without task boundary information.

In CIL, each class is learned only once per task, with all class data available for learning, limiting further class adaptation when data distributions shift, particularly with new domains. Additionally, most CIL methods assume balanced data distributions across classes and tasks and are benchmarked using single-domain datasets like Cifar and mini-ImageNet. However, real-world data streams are typically non-stationary and imbalanced [14] [15].

[16] present a novel approach to quantifying dataset distribution shifts across two dimensions. Their analysis reveals that datasets such as ImageNet [17] and Cifar [18] primarily exhibit correlation shifts in the relationship between features and labels. Conversely, datasets like PACS [19] and DomainNet [20] exemplify diversity shifts, where new features emerge during testing.

We investigate a broader Continual Learning (CL) framework known as task-free online CL (OCL), where data is streamed continuously without defined task boundaries [21], mirroring the non-stationary nature of real-world data. This setup results in imbalances in class and domain distributions, with varying sample availability and domain representation in each batch. As a result, there is a need for continual adjustment of class and data representations to handle the diversity and overlap of class boundaries, especially with the introduction of new class or domain data.

Previous research [5], [7], [9], [22] suggests that rehearsal-based methods are effective in mitigating catastrophic forgetting across various CL scenarios by using a memory set for data replay. This approach is crucial for maintaining CL efficiency in dynamic, imbalanced data conditions. However, existing methods often rely on basic selection strategies, such as random [5] or herding-based sampling [11]. These strategies do not account for imbalanced data distributions and fail to address the increasing intra-class diversity and decreasing inter-class boundaries that occur as new domain and class data are introduced, as illustrated in Fig. 1 (a). Consequently, they are unable to adapt previously acquired knowledge to novel data streams, which require the evolution of learned class

boundaries.

In this paper, we argue that not all streamed data samples are equally beneficial for preserving and enhancing prior knowledge. The most valuable samples often capture the evolving diversity within classes and the similarities between them. To leverage this, we introduce a novel memory-based online CL approach called MSCL. This method has two core features: 1) **Dynamic Memory Adaptation**: MSCL selects samples from incoming data streams that best represent the diversity within classes and the similarities between different classes. To achieve this, we developed the Feature-Distance Based Sample Selection (**FDBS**). FDBS calculates an importance weight for each new sample based on its representational significance compared to the memory set. Especially in imbalanced datasets, our method emphasizes diverse samples within each class and similar samples across different classes, ensuring adaptation of a comprehensive memory set. 2) **Enhanced Data Representation Consolidation with Contrastive Learning**: We introduce a novel contrastive learning loss, denoted as **SCL**, which comprises two key components. The first one is the standard supervised contrastive learning loss **SUP** [23]. This component employs various data augmentations on the original images and enhances representation learning by evaluating the similarity between instances within the current batch. The second one is **IWL** which utilizes the importance weight from FDBS, allowing it to incorporate the similarity between the memory set and the current batch. This approach effectively brings similar class instances closer together while distancing different class instances, thereby enhancing data representation consolidation in the presence of a non-stationary data stream.

The proposed MSCL implements reminiscence of memory plasticity [24], [25] through its Dynamic Memory Adaptation and Enhanced Data Representation Consolidation. By selectively choosing samples that best represent class diversity and similarities, MSCL continuously adapts and updates the memory set, ensuring it remains comprehensive and relevant. Additionally, the contrastive learning loss (SCL) consolidates data representation, facilitating the incorporation of new information and preserving essential knowledge, akin to how human memory plasticity operates.

Our contributions are threefold:

- We design benchmarks for the problem of task free online CL with respect to imbalanced data both in terms of classes and domains, and highlight the limitations of existing CL methods in handling such complex non-stationary data.
- We introduce a novel replay-based online CL method, namely **MSCL**, based on: 1) a novel memory selection strategy, **FDBS** for memeory adaptation, and 2) a novel data importance weight-based Contrastive Learning Loss, **SCL**, for consolidation of data representation.
- The proposed online CL method, **MSCL**, has been rigorously tested across various datasets and architectures, demonstrating superior performance over state-of-the-art memory-based CL methods. It excels particularly in challenging scenarios with imbalanced classes, domains, and combined imbalances. Additionally, we show the versatility of the proposed **MSCL** which can easily integrate with existing CL methods, significantly enhancing their performance.

Preliminary results appeared in [26]. In this paper, we have significantly enhanced our method by incorporating a projection head, employing data augmentation techniques, and extending our previous contrastive loss with a supervised contrastive learning loss (**SUP**). Additionally, we have conducted further detailed ablation studies to highlight the importance of each major design choice.

This paper is organized as follows. Sec. II overviews the related work. Sec. III defines the problem statement. Sec. IV describes our method. Sec. V discusses the experimental settings and results. Sec. VI details the ablation studies. Sec. VII concludes the paper.

II. Related Work

A. Task-Free online continual learning

[5], [21] introduce a novel CL scenario where task boundaries are not predefined, and the model encounters data in an online setting. Several memory-based strategies have been proposed to navigate this scenario. Reservoir Sampling (**ER**) [5] assigns an equal chance for each piece of data to be selected in an online setting. However, this method can be easily biased by imbalanced data stream in terms of class and/or domain and inadvertently miss data that are more representative. Maximally Interfered Retrieval (**MIR**) [6] makes use of **ER** for data selection but retrieves the samples from the memory set which are most interfered for current learning. Gradient-based Sample Selection (**GSS**) [7] proposes to maximize the variance of gradient directions of the data samples in the replay buffer for data sample diversity but with no guarantee that the selected data are class representative. Furthermore, the replay buffer can be quickly saturated without any further update when local maximum of gradient variance is achieved. Online Corset Selection (**OCS**) [8] also employs the model's gradients for cosine similarity computation to select informative and diverse data samples in affinity with past tasks. Unfortunately, they are not class aware and its effectiveness diminishes when handling imbalanced data. In contrast, our proposed MSCL makes use of FDBS to promote the selection of informative data samples in terms of intra-class diversity and inter-class similarity in the feature space for storage. It further improves discriminative data representation using a built-in contrastive loss **SCL**.

B. Imbalanced continual learning

[14] highlighted the limitations of existing CL methods, such as iCaRL [11], in handling numerous classes. The authors attributed these shortcomings to the presence of imbalanced data and an increase in inter-class similarity. To address this, they proposed evaluating CL methods in an imbalanced class-incremental learning scenario, where the data distribution across classes varies ((also known as Long-Tailed Class Incremental Learning, as defined by [15])). In order to mitigate this issue, they introduced a simple bias correction layer to adjust

the final output during testing. One approach described by [22] is CBRS (Class-Balancing Reservoir Sampling), which is based on the reservoir sampling technique [5]. This algorithm assumes equal data storage for each category and employs reservoir sampling within each category. However, when faced with imbalanced domain-incremental learning scenarios where the data distribution within domains is uneven, CBRS can only perform random selection, limiting its effectiveness. Instead, our proposed MSCL performs dynamically class informed data sample selection.

a) Contrastive learning in Continual learning: Continual learning methods(*e.g.*, [27]–[29]) utilizing contrastive learning primarily rely on supervised contrastive learning proposed by [23]. These methods typically necessitate extensive data augmentation to enhance representation learning, yet they often neglect the memory selection process. In our method, we introduce a novel contrastive learning loss. Compared to the standard supervised contrastive learning loss, it exhibits two main differences: 1) It evaluates not only the similarities within the current batch but also between the memory set and the current batch. 2) The loss functions as an adversarial process against our memory selection method, helping to create a more compact feature space.

III. Preliminary and problem statement

We consider the setting of online task-free continual learning. The learner receives non-stationary data stream $\mathbb{O}$ through a series of data batches denoted as $\mathbb{S}_t^{str} = (x_i, y_i)_{i=1}^{N_b}$ at time step t. Here, (x_i, y_i) represents an input data and its label, respectively, and N_b denotes the batch size. The learner is represented as $f(\cdot;\boldsymbol{\theta}) = g \circ F$, where g represents a classifier and F denotes a feature extractor. We define a memory set as $\mathbb{S}^{mem} = (x_j, y_j)_{j=1}^{M}$, where M is the memory size. We use the function $l(\cdot,\cdot)$ to denote the loss function. The global objective from time step 0 to T can be computed as follows:

$$l^* = \sum_{t=0}^{T} \sum_{(x_i,y_i)\in\mathbb{S}_t^{str}} l(f(x_i;\boldsymbol{\theta}), y_i) \tag{1}$$

However, within the setting of online continual learning, the learner does not have access to the entire data at each training step but only the current data batch and those in the memory set if any memory. Therefore, the objective at time step T can be formulated as follows:

$$l_T = \underbrace{\sum_{\mathbb{S}_T^{str}} l(f(x_i;\boldsymbol{\theta}_{T-1}), y_i)}_{\text{current loss}} + \underbrace{\sum_{\mathbb{S}^{mem}} l(f(x_j;\boldsymbol{\theta}_{T-1}), y_j)}_{\text{replay loss}} \tag{2}$$

As a result, to enable online continual learning without catastrophic forgetting, one needs to minimize the gap between l^* and l^T:

$$\min(l^* - l_T) = \min(\sum_{t=0}^{T-1} \sum_{\mathbb{S}_t^{str}\setminus\mathbb{S}^{mem}} l(f(x_i;\boldsymbol{\theta}_{T-1}), y_i)) \tag{3}$$

In this paper, we are interested in memory-based online CL. Our objective is to define a strategy which carefully selects data samples to store in the memory set and continuously refines data representation to minimize the gap as shown in Eq. (3).

IV. Methodology

The proposed method, denoted as **MSCL**, consists of two main components, namely Feature-distance based sample selection (FDBS) (sect.IV-A) and contrastive learning for better discriminative feature representation (sect.IV-B). The whole algorithm is sketched in algo.1.

A. Feature-Distance based sample selection

In the context of imbalanced online domain and class continual learning scenarios, models need to contend with at least two types of distribution shifts: correlation shift and diversity shift. In classification problems, these distribution shifts can result in increased inter-class similarity and intra-class variance, ultimately leading to catastrophic forgetting. Current memory selection methods (e.g., ER [5], CBRS [22], GSS [7], OCS [8]) are unable to effectively address both of these challenges simultaneously. To tackle this issue, we introduce our feature-based method, referred to as Feature-Based Dissimilarity Selection (FDBS). FDBS encourages the model to select data points that are the most dissimilar within a class and the most similar between different classes. This strategy aims to enhance both inter-class similarity and intra-class variance within the memory set. Consequently, FDBS helps to narrow the gap between the memory set and the true data distribution, as highlighted in Equation 3.

Let M to denote the memory size and K the number of data samples so far streamed. Let p to denote our projection head. The current batch size is set to N_b, and the sampled memory batch size is N_m. When the learner receives a batch of data $\mathbb{S}^{str}$ from the stream $\mathbb{O}$, we check for each new data sample x_i in $\mathbb{S}^{str}$ whether the memory set is full. If it is not full, we can directly store x_i. However, if the memory set is full, we need to evaluate the importance weight w_i of the new data sample x_i to determine whether it is worth storing. The key to this process is to keep the memory set aware of intra-class diversity and inter-class boundaries based on the feature distances between the new data sample x_i and the memory set. It involves the following three main steps:

- Sample a batch of data, denoted as $\mathbb{S}^m$, from the memory set with size N_m. Double views the current batch and the memory batch. $\mathbb{S}_{doub}^m$ contains both the original images and the augmented views from the memory batch. Apply the same notation to $\mathbb{S}_{doub}^{str}$. To get the features, we use $z(x) = p \circ F(x)$.
- We then calculate the feature distance, denoted as $\boldsymbol{D}$ (refer to Eq. (4)), between every data point in the set $\mathbb{S}_{doub}^{str}$ and each data sample stored in $\mathbb{S}_{doub}^m$. Subsequently, we identify the minimum distance between the input data and the memory set for each input data sample, resulting in the vector $\boldsymbol{d}^{str}$ as defined in Eq. (4)

$$\boldsymbol{D}_{i,j} = dist\left\{z(x_i), z(x_j)\right\}_{(x_i\in\mathbb{S}_{doub}^{str}; x_j\in\mathbb{S}_{doub}^m)} \tag{4}$$

- Subsequently, we compute $\boldsymbol{D}^{mem}$, as in Eq. (5), the feature distance between every data in $\mathbb{S}_{doub}^m$ and $\mathbb{S}^{mem}$, and the minimum distance for each data point in the memory set in $\boldsymbol{d}^{mem}$, as shown in Eq. (5). We then calculate $\boldsymbol{a}$ as in Eq. (7) a weighted average distance

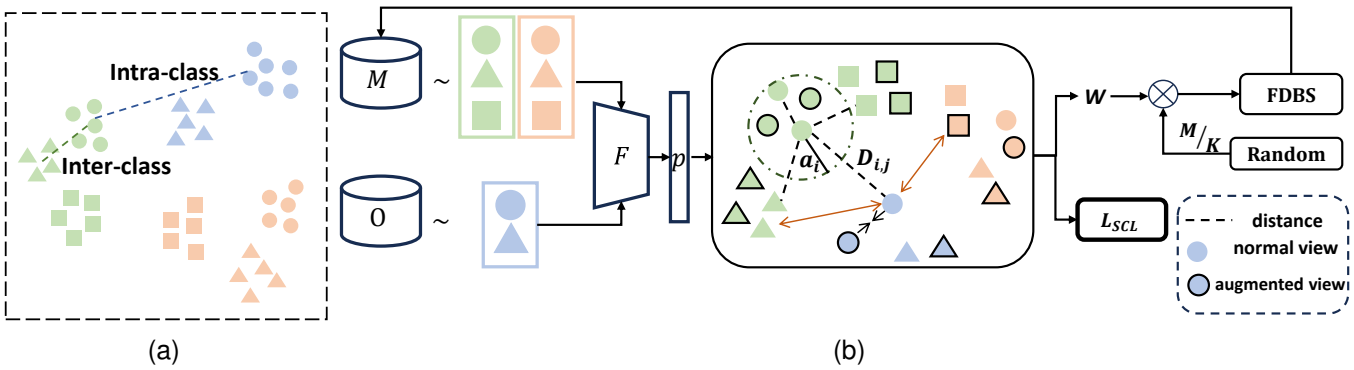

Fig. 1. Both figures illustrate domains using colors and categories with shapes. **(a)** Shows models adapting to datasets with high inter-class similarity and intra-class variance, highlighting the challenge of differentiating closely related categories.**(b)**Our proposed MSCL involves mapping input data and a memory set into a shared feature space. Here, $\boldsymbol{D}_{i,j}$ represents the distance between input data x_i and data x_j in the memory set. We use the same indexing convention for other formulas. We calculate distances, $\boldsymbol{D}$ and $\boldsymbol{a}$, between input data and memory set, and then derive an importance weight matrix quantifying each input data representative importance w.r.t those in the memory set based on the analysis of their intra-class diversity or inter-class similarity in the feature space. These importance weights are combined with random selection to give birth to our Feature-Distance based Sample Selection (FDBS) which identifies the most representative input data points for storage into the memory set. Armed with this importance weight matrix, we proceed to craft a novel Contrastive Loss (SCL) aimed at refining the feature space by compacting intra-class data and creating greater separation among inter-class data.

from a data point in the memory set to all other points, using a RBF kernel as in Eq. (7) to weight the distances. We aim to assign higher weight to closer distances.

$$\boldsymbol{D}_{i,j}^{mem} = dist\left\{z(x_i), z(x_j)\right\}_{(x_i \in \mathbb{S}_{doub}^{m}, x_j \in \mathbb{S}^{mem})} \tag{5}$$

$$\boldsymbol{d}_i^{str} = min(\boldsymbol{D}_{i,:}); \boldsymbol{d}_i^{mem} = min(\boldsymbol{D}_{i,j\neq i}^{mem}) \tag{6}$$

- By computing the difference between $\boldsymbol{a}$ and $\boldsymbol{D}$, we can derive an **importance weight** for each new data. This weight is subsequently combined with the reservoir sampling coefficient to determine the probability of selecting the new data point.

$$\boldsymbol{\alpha}_{i,j} = e^{-\frac{\left\|\boldsymbol{D}_{i,j}^{mem} - \boldsymbol{d}_i^{mem}\right\|^2}{2\sigma^2}}; \boldsymbol{a}_i = \frac{\sum_{j\neq i}^{M} \boldsymbol{D}_{i,j}^{mem} \boldsymbol{\alpha}_{i,j}}{\sum_{j\neq i}^{M} \boldsymbol{\alpha}_{i,j}} \tag{7}$$

Importance weight is the core concept of our proposed method. It serves to assess the significance of a new data sample with respect to the memory set, with a focus on promoting diversity among previously encountered intra-class data while also considering the potential closeness to inter-class boundaries. Specifically, we calculate this importance weight, as defined in Eq. (9), to capture the influence of each data point in the memory set on an input data sample. This influence is determined by whether they belong to the same class, as illustrated in Fig. 1 (b). Our approach is based on the intuitive notion that when two points, x_i and x_j, are closer in proximity, the impact of x_j on x_i becomes more pronounced. To achieve this, we employ a Radial Basis Function (RBF) kernel, as expressed in Eq. (8). This kernel ensures that the influence of distant points diminishes rapidly. Additionally, we use the sign function, as shown in Eq. (8), to assign a value of 1 if the classes are the same and -1 otherwise.

When comparing a new data sample x_i with a memory set data point x_j, we consider two scenarios based on their class labels. If they share the **same class label**, as shown in Fig. 1 (b), and if the feature distance $\boldsymbol{D}_{i,j}$ significantly exceeds $\boldsymbol{a}_j$, it implies a substantial difference between x_i and x_j. In this case, we assign $\boldsymbol{W}_{i,j}$ a value greater than 1, promoting the selection of x_i for storage. However, when x_i and x_j have **different class labels**, we aim to store data points near decision boundaries to capture closer class boundaries caused by increased inter-class similarities. We achieve this by setting $\boldsymbol{W}_{i,j}$ using Eq. (9) with the sign function returning -1. If $\boldsymbol{a}_j$ significantly surpasses $\boldsymbol{D}_{i,j}$, it implies that despite their different labels, x_i closely resembles x_j, motivating us to store x_i. Conversely, if $\boldsymbol{a}_j$ is substantially smaller than $\boldsymbol{D}_{i,j}$, it suggests that the model can readily distinguish between x_i and x_j, leading us to exclude x_i from storage. When $\boldsymbol{D}_{i,j}$ is approximately equal to $\boldsymbol{a}_j$, we consider x_i as a typical data point close to x_j, leading $\boldsymbol{W}_{i,j}$ to approach 1, resulting in a random selection.

$$\boldsymbol{\beta}_{i,j} = e^{-\frac{\|\boldsymbol{D}_{i,j} - \boldsymbol{d}_i^{str}\|^2}{2\tau^2}}; sgn(y_i, y_j) = \begin{cases} 1 \text{ if } y_i = y_j \\ -1 \text{ if } y_i \neq y_j \end{cases} \tag{8}$$

$$\boldsymbol{W}_{i,j} = e^{sgn(y_i,y_j)\frac{\boldsymbol{D}_{i,j} - \boldsymbol{a}_j}{\boldsymbol{D}_{i,j} + \boldsymbol{a}_j}\boldsymbol{\beta}_{i,j}} (y_i \in \mathbb{S}_{doub}^{str}; y_j \in \mathbb{S}_{doub}^{m}) \tag{9}$$

To take into account the influence of all data points in the memory set on a new input data point for its importance weight, we directly multiply the impact of each memory point as shown in Eq. (10).

To get the final probability p_i for a new data sample x_i to be chosen for storage in memory, we introduce the reservoir sampling. Given a fixed memory size M and the number of data samples observed so far in the data stream, denoted as K, M/K represents the probability of each data sample being randomly selected. We then use the importance weight $\boldsymbol{w}_i$ to adjust the probability of the new data sampled x_i being selected, as shown in Eq. (10). This allows us to handle imbalanced data and retain a certain level of randomness.

$$\boldsymbol{w}_i = \frac{\sum_{j=1}^{2N_m} \boldsymbol{W}_{i,j}}{2N_m} \quad ; \quad p_i = min(\boldsymbol{w}_i \frac{M}{K}, 1) \tag{10}$$

B. Contrastive learning for better discriminative feature representation

Our Feature-Distance Based Sample Selection (FDBS) can effectively store the most representative samples during training. However, the latent space of our memory set may not be compact, potentially degrading our classification performance. To address this issue, we introduce the use of contrastive learning loss. Previous methods, such as OnPro [30] and CaSSLe [31], have already employed supervised contrastive learning [23] to learn instance-wise representations:

$$\begin{aligned} L_{\text{SUP}} = & \sum_{i=1}^{2N_b} \frac{1}{|I_i^b|} \sum_{j \in I_i^b} \log \left(\frac{\exp(\text{sim}(z_i^b, z_j^b)/\tau_{sc})}{\sum_{k \neq i} \exp(\text{sim}(z_i^b, z_k^b)/\tau_{sc})} \right) \\ & + \sum_{i=1}^{2N_m} \frac{1}{|I_i^m|} \sum_{j \in I_i^b} \log \left(\frac{\exp(\text{sim}(z_i^m, z_j^m)/\tau_{sc})}{\sum_{k \neq i} \exp(\text{sim}(z_i^m, z_k^m)/\tau_{sc})} \right) \end{aligned} \tag{11}$$

Where N_b and N_m represent the number of training data in the current batch and the batch sampled from the memory set, respectively. I_i is the set of positive samples for z_i. This equation separately computes the supervised contrastive loss for current data and data from the memory set. However, it overlooks the distance between the memory set and current data. To address this issue, we propose the use of an importance weight to compute a specific contrastive learning loss.

The importance weight $\boldsymbol{W}_{i,j}$, derived from Eq. (9), measures feature space similarity between data points and is differentiable. Inspired by contrastive learning's goal to distinguish between similar (positive) and dissimilar (negative) sample pairs. IWL aims to decrease inter-class similarity and intra-class variance, serving as an adversarial element to memory selection and compacting the feature space for better memory selection. For a data batch of size N_b, we select a minibatch from the memory set of size N_m, and compute L_{IWL} as per Eq. (12), optimizing $\boldsymbol{W}_{i,j}$ to align data points with matching class labels closer and separate those with differing labels.

$$L_{IWL} = \frac{\sum_{i=1}^{2} N_m \sum_{j=1}^{2N_b} log(\boldsymbol{W}_{i,j})}{\sum_{i=1}^{2N_m} \sum_{j=1}^{2N_b} \boldsymbol{\beta}_{i,j}} \tag{12}$$

Thus, our total contrastive learning loss is:

$$L_{SCL} = L_{SUP} + L_{IWL} \tag{13}$$

V. Experiments and Results

We introduce balanced CL benchmarks in sect.V-A, define imbalanced ones in sect.V-B, describe the baselines and implementations details in sect.V-C, and present the experimental results both on balanced scenarios in sect.V-D and imbalanced ones in sect.V-E.

Algorithm 1 Train a batch at time step t

Input: F, g ,$\mathbb{S}^{mem}$, $\mathbb{S}^{str}$, n, K, $\mathbf{Z}^{mem}$ stores the features of the memory set, N_b is the current batch size, and N_m is the memory batch size.

for n steps **do**
- sample batch $I, \mathbf{X}^m, \boldsymbol{y}^m$ of size N_m from $\mathbb{S}^{mem}$ $\{I$: the index of the samples in $S^{mem}\}$
- $\mathbf{X}^{str}, \boldsymbol{y}^{str} = \mathbb{S}^{str}$
- $\mathbf{X}^m_{doub} = cat(aug(\mathbf{X}^m), \mathbf{X}^m)$
- $\mathbf{X}^{str}_{doub} = cat(aug(\mathbf{X}^{str}), \mathbf{X}^{str})$
- $\mathbf{Z}^m, \hat{\boldsymbol{y}}^m = p \circ F(\mathbf{X}^m_{doub}), g \circ F(\mathbf{X}^m_{doub})$
- $\mathbf{Z}^{str}, \hat{\boldsymbol{y}}^{str} = p \circ F(\mathbf{X}^{str}_{doub}), g \circ F(\mathbf{X}^{str}_{doub})$
- $\alpha = 0.1 + 0.9 * 0.99^t$
- Current Loss : $L_{cur} = \ell(\hat{\boldsymbol{y}}^{str}, \boldsymbol{y}^{str})$
- Replay Loss : $L_r = \ell(\hat{\boldsymbol{y}}^m, \boldsymbol{y}^m)$
- Update $\mathbf{Z}^{mem}[I] = \mathbf{Z}^m[: N_m]$
- $\boldsymbol{D}^{mem} = dist(\mathbf{Z}^m, \mathbf{Z}^{mem})$ as Eq. (5)
- Compute $\boldsymbol{a}$ based on Eq. (7)
- $\boldsymbol{D} = dist(\mathbf{Z}^{str}, \mathbf{Z}^m)$ as Eq. (4)
- Compute $\boldsymbol{w}$ based on Eq. (9) and Eq. (10)
- $L_{IWL} = L_{IWL}(\boldsymbol{w})$ as Eq. (12)
- $L_{SUP} = L_{SUP}(\mathbf{X}^{str}, \boldsymbol{y}^{str}, \mathbf{X}^{mem}, \boldsymbol{y}^{mem})$ as Eq. (11)
- Total Loss : $L = \alpha L_{cur} + (1-\alpha) L_r + L_{IWL} + L_{SUP}$
- Update: F, g : Adam.step( )
- FDBS($\mathbb{S}^{mem}$,$\mathbb{S}^{str}_t$,$\boldsymbol{w}$,$\boldsymbol{D}$,M,K,$\mathbf{Z}^{mem}$) as shown in Algorithm 2

end for

A. Balanced benchmarks

Building upon previous research [7], [9], [13], we utilize three well-established Continual Learning (CL) benchmarks: Split Mini-ImageNet, Split CIFAR-100, and PACS. For Split CIFAR-100, we partition the original CIFAR-100 dataset [18] into ten subsets, with each subset representing a distinct task comprising ten classes. For Split Mini-ImageNet [17], we partition the original Mini-ImageNet dataset [18] into 10 subsets, with each subset representing a distinct task comprising ten classes. As for PACS [19], it encompasses four domains: photo, art painting, cartoon, and sketch. Each domain consists of the same seven classes. In our experiments, we treat each domain as an individual task, resulting in a total of four tasks. Notably, due to significant differences between images in each domain, one can observe a notable increase in inter-class variance within this dataset.

B. Imbalanced benchmarks

Existing CL benchmarks, with uniform class and domain distributions, fail to test CL methods on non-stationary, imbalanced data. Thus, we've created benchmarks specifically to assess CL methods' robustness to data imbalance.

1) Imbalanced Class-Incremental Learning (Imb CIL).: To establish an imbalanced Class-incremental scenario for split CIFAR-100 and split mini-ImageNet, we build upon the approach introduced by [22]. Unlike traditional benchmarks that distribute instances equally among classes, we induce

Algorithm 2 FDBS at time step t

Input: $\mathbb{S}^{mem}$, $\mathbb{S}^{str}$, $\boldsymbol{w}$, $\boldsymbol{D}$, M, K, $\mathbf{Z}^{mem}$

$\mathbf{X}^{mem}, \boldsymbol{y}^{mem} = \mathbb{S}^{mem}$;
for each data $i, (x_i, y_i)$ in $\mathbb{S}_t^{str}$ **do**
  $K = K + 1$
  if $len(\mathbb{S}^{mem}) < M$ **then**
    store (x_i, y_i) in $\mathbb{S}^{mem}$
  else
    $p = \boldsymbol{w}_i * M/K$
    $r = random.rand()$
    if $r < p$ or $y_i \notin \mathbb{S}^{mem}$ **then**
      $c = most\_frequent(\boldsymbol{y}^{mem})$
      $I = index(\boldsymbol{y}^{mem} == c)$
      $k = random.choice(I)$
      $\mathbf{X}^{mem}[k], \boldsymbol{y}^{mem}[k] = x_i, y_i$;
      $\mathbf{Z}^{mem}[k] = Z(x_i)$
    else
      ignore (x_i, y_i)
    end if
  end if
end for

class imbalance by utilizing a predefined ratio vector, denoted as **r**, encompassing five distinct ratios: $(10^{-2}, 10^{-1.5}, 10^{-1}, 10^{-0.5}, 10^{0})$. In this setup, for each run and each class, we randomly select a ratio from **r** and multiply it by the number of images corresponding to that class. This calculation determines the final number of images allocated to the class, thus establishing our imbalanced class scenario. We maintain the remaining conditions consistent with the corresponding balanced scenario.

2) Imbalanced Domain-incremental Learning (Imb DIL): We adapt the PACS dataset, encompassing four domains, and follow an approach akin to our Imbalanced Class-Incremental method. For each domain, we randomly select a ratio from **r**, multiply it with the image count of the domain, thereby maintaining a balanced class count within the imbalanced domain.

3) Imbalanced Class and Domain Incremental Learning (Imb C-DIL).: We further refine the PACS dataset to generate an imbalanced class-domain incremental scenario, which mirrors a more realistic data setting. This scenario involves randomly selecting a ratio from **r** for each class and domain, and multiplying it with the count of instances for that class within the domain. This operation yields $4*7$ values for PACS, resulting in a diverse number of data points across different classes and domains. This approach accentuates the growth of inter-class similarity and intra-class variance. Because both the class and domain are already imbalanced in the original **DomainNet** [20], we directly use its original format to generate the imbalanced scenario. We adhere to a sampling without replacement strategy for data stream generation. Once data from a pair of class and domain is exhausted, we transition to the next pair.

C. Baselines and implementation details

As the proposed FDBS is a memory-based online CL method, we compare it primarily against other memory-centric techniques such as Experience Replay (ER) [5], Gradient-Based Sample Selection (GSS) [7], Class-Balancing Reservoir Sampling (CBRS) [22], Maximally Interfering Retrieval (MIR) [6], and Online Corset Selection(OCS) [8]. Online Prototype Learning(OnPro) [30] achieved the SOTA performance in the Class-incremental scenario over Cifar-100 and Mini-ImageNet.

We compare Fine-tuning (F.T.), where pre-existing model parameters are used as starting points for new tasks without additional data, against i.i.d. offline training, a method that grants complete access to the dataset, allowing multiple data reviews for maximum performance. In this comparison, FT represents the lower bound of performance, while offline training serves as the upper bound. Our method introduces Feature-Distance Based Sampling (FDBS) for choosing samples and Contrastive Learning Loss for better representation learning. We test the effectiveness of FDBS combined with L_{SCL} in our experiments.

We adopt a reduced ResNet-18 architecture similar to that used in [5]. We maintain a fixed batch size of 20 for the incoming data stream, with one update steps per batch. We set the σ value in our radial basis function (RBF) kernel at 0.1, and the τ value in Eq. (9) at 1.0. Our approach's performance is evaluated across the balanced and imbalanced benchmarks through five independent runs, from which we compute the average accuracy.

D. Results on balanced benchmarks

Results for balanced scenarios are shown in Tab. I. In class incremental learning (CIL) scenarios such as split Mini-ImageNet and CIFAR-100, classical methods like ER, CBRS, and GSS do not perform well with low memory sizes. This is because, with a low memory size relative to the training data size, these methods heavily bias towards the memory data. As the memory size increases, the performance of these methods significantly improves. OnPro, which uses rich data augmentation and evaluates the class mean for each update, performs very well in these scenarios. In comparison, our method uses a more representative selection strategy and incorporates a comprehensive contrastive loss, leading to consistent improvements in results. In domain incremental learning (DIL) scenarios such as PACS, OnPro does not perform as well as in CIL scenarios, because the class mean loses its significance across multiple domains. However, our method still achieves the best results. Our memory selection strategy aims to increase intra-class variance, leading to greater diversity in the memory and improved performance. Additionally, MSCL maintains stable performance with lower standard deviations, indicating more reliable and consistent results. This robustness, combined with its superior accuracy, highlights MSCL's efficiency and reliability in handling various memory sizes and datasets.

TABLE I
WE REPORT THE RESULTS OF OUR EXPERIMENTS CONDUCTED ON **BALANCED** SCENARIOS. WE PRESENT THE FINAL ACCURACY AS MEAN AND STANDARD DEVIATION OVER FIVE INDEPENDENT RUNS.

	Mini-ImageNet			CIFAR-100			PACS		
F.T.	4.2 ± 0.2			4.4±0.2			20.6±0.2		
i.i.d. Off	52.5 ± 0.1			49.8±0.3			59.6±0.1		
	M=1k	M=2k	M=5k	M=1k	M=2k	M=5k	M=0.1k	M=0.2k	M=0.5k
ER	10.1±0.7	13.2±0.8	16.5±1.8	11.0±0.7	14.2±0.5	20.2±0.9	36.1±1.2	38.6±1.4	39.8±1.5
GSS	10.2±0.6	13.1 ± 1.2	14.2±0.9	10.3 ± 0.5	13.3 ± 0.5	17.5 ± 1.2	35.8 ± 2.8	37.8 ± 3.2	38.7 ± 2.2
CBRS	10.3±0.8	13.5 ± 0.9	16.4±2.1	11.0 ± 0.6	14.5 ± 0.8	20.5± 0.8	36.3 ± 1.1	38.8 ± 1.6	40.1 ± 1.7
MIR	10.7±0.7	14.8 ± 1.1	17.5±1.5	11.5 ± 0.4	15.1 ± 0.5	21.7 ± 0.9	37.6 ± 0.9	40.2 ± 0.8	43.2 ± 1.2
OCS	10.8±0.5	15.1 ± 1.1	17.8±1.6	11.4 ± 0.5	14.8 ± 0.8	21.3 ± 0.9	36.8 ± 0.7	39.6 ± 0.7	42.2 ± 1.1
OnPro	21.2±0.4	30.5 ± 0.5	34.5±0.8	26.6 ± 0.5	30.6 ± 0.8	36.6 ± 0.8	36.3 ± 1.3	40.5 ± 1.3	41.4 ± 1.5
MSCL(ours)	**24.7**±0.4	**33.9** ± 0.5	**36.9**±0.9	**27.5** ± 0.4	**31.2** ± 0.7	**37.5** ± 0.8	**38.8** ± 0.9	**42.7** ± 1.1	**45.8** ± 1.3

TABLE II
RESULTS ON OUR **IMBALANCED** SCENARIOS. WE PRESENT THE FINAL ACCURACY AS MEAN AND STANDARD DEVIATION OVER FIVE INDEPENDENT RUNS. FOR PACS, THE MEMORY SIZE WAS SET TO 1000, WHILE FOR ALL OTHER SCENARIOS, THE MEMORY SIZE WAS SET TO 5000.

Scenarios	**Imb CIL**		**Imb DIL**	**Imb C-DIL**	
	CIFAR-100	Mini-ImageNet	PACS	PACS	DomainNet
Fine Tunning	3.1 ± 0.3	3.5 ± 0.2	15.5 ± 1.3	14.3 ± 1.2	2.3 ± 0.6
i.i.d. Offline	41.6 ± 0.5	43.1 ± 0.6	46.3 ± 0.4	46.1 ± 0.9	37.2 ± 0.7
ER	7.1 ± 0.8	8.2 ± 1.3	25.6 ± 2.1	22.4 ± 1.3	6.2 ± 0.6
GSS	8.3 ± 0.7	7.9 ± 0.5	24.4 ± 1.7	20.2 ± 2.1	5.1 ± 0.4
CBRS	10.2 ± 0.4	11.3 ± 0.6	25.9 ± 1.5	23.6 ± 1.7	6.1 ± 0.6
MIR	7.5 ± 0.9	8.9 ± 0.3	25.8 ± 2.1	22.2 ± 2.5	6.4 ± 0.4
OCS	11.6 ± 0.6	12.3 ± 0.4	27.1 ± 1.4	24.7 ± 1.3	8.4 ± 0.7
OnPro	22.3 ± 0.5	15.8 ± 0.7	27.1 ± 1.7	25.5 ± 1.4	11.2 ± 0.9
MSCL(Ours)	**24.8**±0.6	**17.2**±0.4	**31.2**±0.8	**30.6**±0.7	**12.4**±0.7

E. Results on imbalanced scenarios

Tab. II displays the experimental results in the imbalanced settings. For imbalanced CIL scenarios, the CBRS method, which maintains an equal count of images from each class in memory, outperforms the basic ER approach. Meanwhile, OCS, by continuously evaluating data batch gradients, filters noise and selects more representative data, shining particularly in imbalanced contexts. However, our method stands out, consistently leading in all imbalanced tests. As scenarios evolve from Imb DIL to Imb C-DIL, other methods' accuracy drops significantly, but FDBS maintains robust performance. Its strength lies in using feature-distance to fine-tune memory selection, preserving class boundaries and boosting intra-class diversity.

VI. ABLATION STUDY AND HYPERPARAMETER ANALYSIS

We conduct an ablation study in Sec. VI-A, discuss the impact of σ and τ of RBF kernel in Sec. VI-B, compare the running time of different methods in Sec. VI-C, assess the impact of memory size in Sec. VI-D, evaluate average forgetting in Sec. VI-E, illustrate the distribution of our memory set in Sec. VI-F, explore the integration of our method with other methods in Sec. VI-G, and finally present the results of the methods in the classical class incremental scenario in Sec. VI-H.

A. Ablation study

Our method comprises two key components: the memory selection method (FDBS) for memory adaptation and the contrastive learning loss L_{SCL}, as detailed in Eq. (13), for evolving data representation consolidation. Tab.VI-A highlights the contributions and effectiveness of each component. As can be seen there, memory adaptation by FDBS and data representation consolidation through L_{SCL} prove to be both useful and complementary, with FDBS consistently enhancing performance, especially in imbalanced scenarios, while L_{SCL} appears further critical.

TABLE III
ABLATION STUDIES ON BALANCED CIFAR-100 AND IMBALANCED DOMAINNET. WE SET THE MEMORY SIZE TO 5000.

Method	Balanced CIFAR-100	Imb DomainNet
F.T.	4.4 ± 0.2	2.3 ± 0.6
w/o L_{SCL}	22.1 ± 1.2	7.8 ± 0.8
w/o FDBS	34.7 ± 0.9	9.5 ± 0.9
MSCL	**37.5** ± 0.8	**12.4** ± 0.7

B. The impact of σ in RBF kernel

The Radial Basis Function (RBF) kernel is a widely used kernel function in machine learning, defined as [32]:

$$K(x_1, x_2) = exp(-\frac{||x_1 - x_2||^2}{2\sigma^2}) \tag{14}$$

In our implementation, we normalize the feature vectors x_1 and x_2 such that $||x_1|| = 1$ and $||x_2|| = 1$. With this normalization, the squared Euclidean distance between x_1 and x_2 satisfies $||x_1 - x_2|| \in [0, 2]$, since the maximum distance occurs when x_1 and x_2 are in opposite directions.

To illustrate the effect of σ on the kernel values, we plot $K(x_1, x_2)$ for different values of σ over the range $||x_1 - x_2|| \in [0, 2]$:

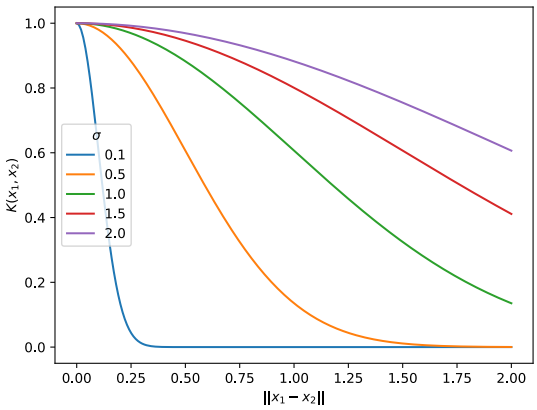

Fig. 2. RBF kernel values with different σ

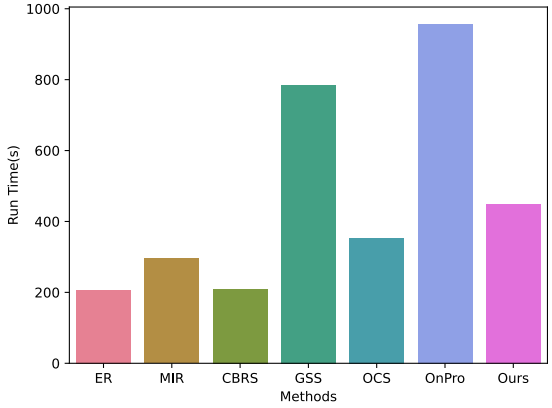

Fig. 3. Running Time of different methods on Blanced CIFAR-100.

$\tau \backslash \sigma$	0.1	0.5	1.0	1.5
0.1	30.5	28.7	28.9	30.1
0.5	29.6	28.2	28.0	28.8
1.0	**31.5**	29.1	29.5	30.6
1.5	30.6	30.3	29.6	30.5

TABLE IV
RESULTS OF VARING σ AND τ ON BLANACED CIFAR-100 WITH A MEMORY SIZE OF 2000.

As shown in Fig. 2, the parameter σ controls the radius of influence of the kernel function. When σ is small, the kernel value $K(x_1, x_2)$ is significant only when x_1 and x_2 are very close; for larger distances, the kernel value approaches zero rapidly. Conversely, a larger σ results in a broader influence, allowing more distant points to contribute meaningfully to the kernel value.

To evaluate the impact of σ and another parameter τ on our method, we conducted experiments using a memory size of 2000 on the Balanced CIFAR-100 dataset. The results are summarized in Tab. IV.

In our framework, σ is the parameter in Eq. (7), which determines the number of points that significantly contribute to the calculation of the average distance a from a data point in the memory set to other points. A smaller σ means that only nearby points have a substantial impact on a, effectively focusing on local neighborhoods.

Similarly, τ is the parameter in Eq. (8), which influences the calculation of the importance weights w. A larger τ allows for contributions from more distant points when computing these weights.

Our experimental results indicate that setting $\sigma = 0.1$ and $\tau = 1.0$ yields the best performance. This suggests that when calculating the average distance a within the memory set, it is beneficial to focus on the nearest points, as distant points may introduce noise or irrelevant information. However, when computing the importance weights w, considering the influence of all points in the memory set (achieved by a larger τ) is advantageous.

C. Running Time

In this section, we evaluate the overall running time of our method on the Balanced CIFAR-100 scenario with a memory size of 5K. The results are presented in Fig. 3. Our method shows only a minor increase in running time compared to ER and MIR, while achieving significantly better performance.

D. The impact of memory size

We compare our FDBS with other memory selection methods by adjusting the size of the memory set. The experiments were conducted using the imbalanced class-domain incremental scenario of PACS, and the results are presented in Tab. V.

The experimental results show consistent performance improvements for our proposed FDBS method across all memory sizes tested. Our method outperforms all other memory selection methods in each case, with the magnitude of the improvement being more pronounced for larger memory sizes.

TABLE V
COMPARISON OF DIFFERENT MEMORY SELECTION METHODS ON IMB C-DIL PACS FOR THREE DIFFERENT MEMORY SIZES. WE PRESENT THE FINAL ACCURACY AS MEAN AND STANDARD DEVIATION OVER FIVE INDEPENDENT RUNS

	Memory size			
Methods	100	200	500	1000
ER	16.4±2.3	18.3±2.5	20.4± 1.8	22.4± 1.3
GSS	15.7±1.6	16.6±1.9	18.2±2.3	20.2±2.1
CBRS	17.2±2.1	19.1±2.1	21.6±1.5	23.6±1.7
OCS	18.3±1.8	21.4±2.2	22.7±1.6	24.7±1.3
FDBS	**19.7**±1.9	**23.5**±2.6	**24.7**±2.0	**26.8**±2.2

E. Comprehensive Evaluation of Average Forgetting

We use the metric known as Average Forgetting [33] to measure the extent of knowledge forgotten after training. We compare our method with different approaches across three typical scenarios: balanced CIFAR-100, imbalanced CIFAR-100, and imbalanced class and domain PACS. For the experiments, we set the memory size to 5k for CIFAR-100 and 1k for PACS.

TABLE VI
COMPARISON AVERAGE FORGETTING OF DIFFERENT METHODS ON BALANCED CIFAR-100, IMBALANCED CIFAR-100, AND IMBALANCED CLASS-DOMAIN PACS. WE PRESENT THE FINAL ACCURACY AS MEAN AND STANDARD DEVIATION OVER FIVE INDEPENDENT RUNS

	CIFAR-100	Imb CIFAR-100	Imb C-DIL PACS
F.T.	53.2 ± 2.8	27.5 ± 1.6	35.5 ± 2.2
ER	40.8 ± 3.5	22.7 ± 1.3	23.9 ± 1.5
GSS	38.2 ± 2.3	23.5 ± 1.8	25.7 ± 1.4
CBRS	37.4 ± 3.1	17.8 ± 1.1	22.8 ± 1.5
MIR	35.6 ± 1.8	22.3 ± 1.5	23.5 ± 1.9
OCS	22.5 ± 1.5	16.5 ± 0.9	21.4 ± 1.4
OnPro	16.3 ± 1.4	14.7 ± 0.9	20.4 ± 1.4
MSCL(ours)	**15.4** ± 1.1	**13.6** ± 0.8	**17.5** ± 0.9

Tab. VI demonstrates that, in both balanced and imbalanced scenarios, our method achieves the lowest forgetting and has a lower standard deviation. This indicates that our method is better at retaining learned knowledge while adapting to new information.

F. The distribution of our memory set

To gain deeper insights into the efficacy of our memory selection method, we examine the distribution of our memory set. Our experiments focus on the challenging task of imbalanced Domain-Incremental Learning using the PACS dataset, which comprises four distinct domains (e.g., photo, art painting, cartoon, and sketch). Following training, we analyze the distribution of our memory set, shedding light on how our method has shaped the representation of critical data points within this dynamic learning environment. The results of this analysis are presented in Tab. VII, while the ratios of different domains within the memory set generated by various methods are shown in Fig. 4.

Methods such as ER and CBRS opt for random image selection, aiming to maintain a distribution akin to the original dataset. In contrast, our method prioritizes increasing intra-class diversity, thereby influencing a more balanced distribution of stored images. This approach plays a crucial role in improving the overall performance of continual learning. Additionally, the integration of our Contrastive Learning Loss (SCL) further enhances the feature space consolidation within our memory set. This refinement proves instrumental in effectively capturing images from minority domains, contributing to a more robust and balanced representation of data.

Methods /Domains	Photo	Art Painting	Cartoon	Sketch
Our Scenario	500	1000	2000	3000
ER	78	155	320	447
GSS	125	570	248	57
CBRS	73	162	342	423
OCS	130	183	286	401
FDBS(ours)	156	193	339	312
MSCL(Ours)	190	227	291	292

TABLE VII
COMPARISON OF MEMORY SET COMPOSITION ACROSS METHODS IN IMBALANCED DOMAIN-INCREMENTAL LEARNING (IMB DIL) SCENARIO OF PACS. WE SET THE MEMORY SIZE AS 1000.

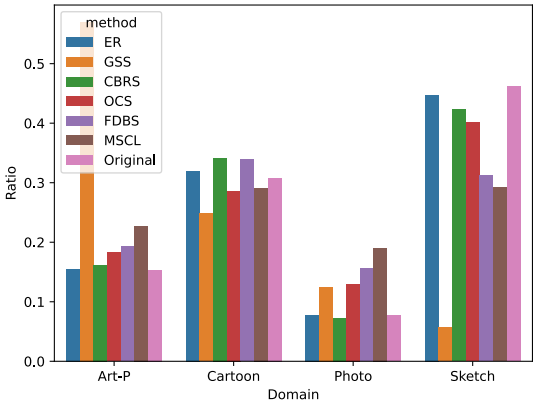

Fig. 4. The ratio of different domains within the memory set compared to the original scenario.

G. Collaborative Learning with other memory-based methods

In our evaluation, we consider three notable continual learning methods, PodNet [13] and AFC [34]. We integrate our Feature-Distance Based Sample Selection (FDBS) method instead of their primary memory selection method, which was originally either random or based on herding. We also introduce our novel contrastive learning loss **SCL**. Our experiments encompass two distinct scenarios: Balanced CIFAR-100 and the imbalanced Class-Domain Incremental Learning (imb C-DIL) of PACS. The results of these experiments are presented in Tab. VIII. Remarkably, our method consistently enhances the performance of these continual learning methods both on balanced and imbalanced scenarios.

Methods	Split-CIFAR100	Imb C-DIL PACS
PodNet	19.5 ± 1.4	20.4 ± 1.1
PodNet + **MSCL**	**25.6** ± 2.3	**29.5**± 0.8
AFC	19.4 ± 1.7	21.5 ± 1.2
AFC + **MSCL**	25.4 ± 2.6	27.6± 0.9

TABLE VIII
COMBINING FDBS WITH OTHER MEMORY-BASED METHODS: EXPERIMENTS ON BALANCED SPLIT CIFAR-100 (MEMORY SIZE: 5000) AND IMBALANCED CLASS-DOMAIN INCREMENTAL LEARNING ON PACS (MEMORY SIZE: 1000).THE FINAL ACCURACY WAS PRESENTED AS THE MEAN AND STANDARD DEVIATION OVER FIVE INDEPENDENT RUNS.

H. Results on Balanced class-incremental learning scenario

We have further evaluated the effectiveness of our proposed approach in the context of classic balanced class-incremental learning. In this scenario, the task boundary is well-defined, and for each task, we employ offline training for multiple epochs. For this purpose, we conducted an experiment named **Cifar 100-B0** as detailed in [35]. In this experiment, we partitioned the original Cifar 100 dataset into 10 and 20 distinct tasks, with each task encompassing a set of 5 distinct classes. The memory size is set as 2000. The result is presented in Tab. IX. Even in the classic class-incremental learning scenario, our proposed method can still significantly improve the previous state-of-the-art method.

Methods	10 steps	20 steps
iCaRL* [11]	65.2 ± 1.0	61.2 ± 0.8
BiC* [14]	68.8 ± 1.2	66.4 ± 0.3
PodNet* [13]	58.0 ± 1.3	53.9 ± 0.8
AFC [34]	61.2 ± 1.4	54.7 ± 0.8
WA* [36]	69.4 ± 0.3	67.3 ± 0.2
MSCL(ours)	**72.5** ± 0.4	**70.5** ± 0.5

TABLE IX
RESULTS FOR CLASSIC CLASS-INCREMENTAL LEARNING ON CIFAR-100. RESULTS MARKED WITH '*' ARE OBTAINED DIRECTLY FROM [35]. THE MEMORY SIZE IS SET TO 2000.

VII. CONCLUSION

This paper presents a new online Continual Learning (CL) method, MSCL, consisted of Feature-Distance Based Sample Selection (FDBS) and Contrastive Learning Loss (SCL). FDBS selects representative examples by evaluating the distance between new and memory-set data, emphasizing dissimilar intra-class and similar inter-class data, thus increasing memory awareness of class diversity and boundaries. SCL minimizes intra-class and maximizes inter-class distances, enhancing discriminative feature representation. Extensive experiments confirmed that FDBS and SCL together outperform other memory-based CL methods in balanced and imbalanced scenarios. Future work will explore combining MSCL with a distillation-based CL method to further improve its performance.

VIII. ACKNOWLEDGMENTS

This work was supported by the French national program of investment of the futur and the regions through the PSPC FAIR Waste project, as well as the French Research Agency, l'Agence Nationale de Recherche (ANR), through the projects Chiron (ANR-20-IADJ-0001-01), Aristotle (ANR-21-FAI1-0009-01), and Astérix (ANR-23-EDIA-0002-001).

REFERENCES

[1] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[2] Friedemann Zenke, Ben Poole, and Surya Ganguli, "Continual learning through synaptic intelligence," in *Proceedings of the 34th International Conference on Machine Learning*, Doina Precup and Yee Whye Teh, Eds., International Convention Centre, Sydney, Australia, 06–11 Aug 2017, vol. 70 of *Proceedings of Machine Learning Research*, pp. 3987–3995, PMLR.

[3] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell, "Progressive neural networks," *CoRR*, vol. abs/1606.04671, 2016.

[4] Vinay Kumar Verma, Kevin J. Liang, Nikhil Mehta, Piyush Rai, and Lawrence Carin, "Efficient feature transformations for discriminative and generative continual learning," *CVPR*, vol. abs/2103.13558, 2021.

[5] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne, "Experience replay for continual learning," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. 2019, vol. 32, Curran Associates, Inc.

[6] Rahaf Aljundi, Lucas Caccia, Eugene Belilovsky, Massimo Caccia, Min Lin, Laurent Charlin, and Tinne Tuytelaars, "Online continual learning with maximally interfered retrieval," 2019.

[7] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio, "Online continual learning with no task boundaries," *CoRR*, vol. abs/1903.08671, 2019.

[8] Jaehong Yoon, Divyam Madaan, Eunho Yang, and Sung Ju Hwang, "Online coreset selection for rehearsal-based continual learning," in *International Conference on Learning Representations*, 2022.

[9] Gido M. van de Ven and Andreas S. Tolias, "Three scenarios for continual learning," *CoRR*, vol. abs/1904.07734, 2019.

[10] Joan Serrà, Dídac Surís, Marius Miron, and Alexandros Karatzoglou, "Overcoming catastrophic forgetting with hard attention to the task," 2018.

[11] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, G. Sperl, and Christoph H. Lampert, "icarl: Incremental classifier and representation learning," *CVPR*, pp. 5533–5542, 2017.

[12] Qiankun Gao, Chen Zhao, Yifan Sun, Teng Xi, Gang Zhang, Bernard Ghanem, and Jian Zhang, "A unified continual learning framework with general parameter-efficient tuning," 2023.

[13] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle, "Podnet: Pooled outputs distillation for small-tasks incremental learning," in *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, 2020.

[14] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu, "Large scale incremental learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 374–382.

[15] Xialei Liu, Yu-Song Hu, Xu-Sheng Cao, Andrew D. Bagdanov, Ke Li, and Ming-Ming Cheng, "Long-tailed class incremental learning," 2022.

[16] Nanyang Ye, Kaican Li, Haoyue Bai, Runpeng Yu, Lanqing Hong, Fengwei Zhou, Zhenguo Li, and Jun Zhu, "Ood-bench: Quantifying and understanding two dimensions of out-of-distribution generalization," 2022.

[17] Oriol Vinyals, Charles Blundell, Timothy P. Lillicrap, Koray Kavukcuoglu, and Daan Wierstra, "Matching networks for one shot learning," *CoRR*, vol. abs/1606.04080, 2016.

[18] Alex Krizhevsky, "Learning multiple layers of features from tiny images," 2009.

[19] D. Li, Y. Yang, Y. Song, and T. M. Hospedales, "Deeper, broader and artier domain generalization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Los Alamitos, CA, USA, oct 2017, pp. 5543–5551, IEEE Computer Society.

[20] 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 International Conference on Computer Vision*, 2019, pp. 1406–1415.

[21] Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars, "Task-free continual learning," 2018.

[22] Aristotelis Chrysakis and Marie-Francine Moens, "Online continual learning from imbalanced data," in *International Conference on Machine Learning*, 2020.

[23] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan, "Supervised contrastive learning," 2021.

[24] Stephen Grossberg, *How Does a Brain Build a Cognitive Code?*, pp. 1–52, Springer Netherlands, Dordrecht, 1982.

[25] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne, "Experience replay for continual learning," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. 2019, vol. 32, Curran Associates, Inc.

[26] Rui Yang, Emmanuel Dellandrea, Matthieu Grard, and Liming Chen, "Imbalanced data robust online continual learning based on evolving class aware memory selection and built-in contrastive representation learning," in *2024 IEEE International Conference on Image Processing (ICIP)*, 2024, pp. 277–283.

[27] Matthias De Lange and Tinne Tuytelaars, "Continual prototype evolution: Learning online from non-stationary data streams," 2021.

[28] Zheda Mai, Ruiwen Li, Hyunwoo Kim, and Scott Sanner, "Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning," 2021.

[29] Yujie Wei, Jiaxin Ye, Zhizhong Huang, Junping Zhang, and Hongming Shan, "Online prototype learning for online continual learning," 2023.

[30] Yujie Wei, Jiaxin Ye, Zhizhong Huang, Junping Zhang, and Hongming Shan, "Online prototype learning for online continual learning," 2023.

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

[32] Jean-Philippe Vert, Koji Tsuda, and Bernhard Schölkopf, *A Primer on Kernel Methods*, pp. 35–70, MITPRESS, 07 2004.
[33] Arslan Chaudhry, Puneet K. Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," in *Computer Vision – ECCV 2018*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, Eds., Cham, 2018, pp. 556–572, Springer International Publishing.
[34] Minsoo Kang, Jaeyoo Park, and Bohyung Han, "Class-Incremental Learning by Knowledge Distillation with Adaptive Feature Consolidation," in *CVPR*, 2022.
[35] Shipeng Yan, Jiangwei Xie, and Xuming He, "Der: Dynamically expandable representation for class incremental learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
[36] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shutao Xia, "Maintaining discrimination and fairness in class incremental learning," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13205–13214, 2019.