Imbalanced data robust online continual learning based on evolving class aware memory selection and built-in contrastive representation learning

Rui Yang, Matthieu Grard, Emmanuel Dellandréa, Liming Chen

IMBALANCED DATA ROBUST ONLINE CONTINUAL LEARNING BASED ON EVOLVING CLASS AWARE MEMORY SELECTION AND BUILT-IN CONTRASTIVE REPRESENTATION LEARNING

Rui YANG[1], Emmanuel Dellandrea[1], Matthieu Grard[2] & 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, 42 Saint-Etienne, France

ABSTRACT

We introduce Memory Selection with Contrastive Learning (MSCL), an advanced Continual Learning (CL) approach, addressing challenges in dynamic and imbalanced environments. MSCL combines Feature-Distance Based Sample Selection (FDBS) for memory management, focusing on inter-class similarities and intra-class diversity, with a contrastive learning loss (IWL) for adaptive data representation. Our evaluations on datasets like MNIST, Cifar-100, mini-ImageNet, PACS, and DomainNet show that MSCL not only competes with but often surpasses existing memory-based CL methods, particularly in imbalanced scenarios, enhancing both balanced and imbalanced learning performance.

Index Terms— Continual Learning, Transfer Learning, Memory Selection

1. INTRODUCTION

Continual Learning (CL) assumes that a model learns from a continuous stream of data over time, without access to previously seen data. It faces the challenge of *catastrophic forgetting*, which occurs when a model forgets previously learned knowledge as it learns new information. State of the art has featured three major CL approaches (*e.g.*, Regularisation-based [1, 2], Parameter isolation oriented [3, 4]) and rehearsal-based [5, 6, 7, 8], along with various CL paradigms [9] (*e.g.*, Task-incremental learning (TIL), Domain-incremental learning (DIL), and Class-incremental learning (CIL)). Early CL methods, *e.g.*, [1, 10], primarily adopted a task-incremental learning (TIL) paradigm and made the unrealistic assumption of having access to task boundaries not only during training for knowledge consolidation but also during inference. As a result, most recent research on CL has focused on class incremental learning (CIL), *e.g.*,[11, 12, 13],which require the model to learn from a sequence of mutually class exclusive tasks and perform the inference without task boundary information. However, in such a scenario, each class can be learned only once within a task with all the class data assumed available for learning and thereby prevents further class adaptation when data distribution shifts for already seen classes come to occur, in particular with new domains. Furthermore, a vast majority of these CIL methods only consider balanced distribution over classes and tasks and are benchmarked using some single domain datasets, *e.g.*, Cifar, mini-ImageNet, although streamed data distributions in CL are generally non-stationary in the real world. As a result, they face significant challenges in the presence of imbalanced data in class and domain [14][15]. [16] introduce a novel approach for quantifying dataset distribution shifts across two distinct dimensions. Their analysis highlights that datasets such as ImageNet[17] and Cifar[18] primarily showcase correlation shifts, characterized by alterations in the relationship between features and labels. In contrast, datasets like PACS[19] and DomainNet[20] predominantly exemplify diversity shifts, marked by the emergence of new features during testing.

We explore a broader Continual Learning (CL) framework, task-free online CL (OCL), where data are streamed without task boundaries[21], reflecting the non-stationary nature of real-life data. This setup leads to imbalances in class and domain distributions, with varying sample availability and domain representation in each batch. Consequently, this necessitates ongoing adjustment of class and data representations to accommodate the diversity and overlap of class boundaries, especially with the introduction of new class or domain data. Prior research(*e.g.*,[5, 9, 22, 7]) indicates rehearsal-based methods excel in addressing catastrophic forgetting across various CL scenarios by using a memory set for data replay, crucially affecting CL efficiency in dynamic, imbalanced data conditions. Yet, existing methods often employ basic selection strategies like random[5] or herding-based sampling[11]. They are unaware of imbalanced data distributions and ignore increasing intra-class diversity and decreasing inter-class boundaries when new domain and/or class data occur over the course of incoming data streams as illustrated in Fig. 1 (a), thereby failing to adapt the previously acquired knowledge to novel data streams which require

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 similarities between them. To harness this, we introduce a novel memory-based online CL approach, MSCL. This method has two core features: 1) **Dynamic Memory Population**: MSCL selects samples from incoming data streams that best represent diversity within classes and similarities between different classes. To achieve this, we've devised 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 a comprehensive memory set. 2) **Enhanced Data Representation with Contrastive Learning**: We've integrated a new Contrastive Learning Loss, **IWL**. This loss uses the importance weight from FDBS to bring similar class instances closer while distancing different class instances. In essence, MSCL continually curates a memory set that captures the dynamic nature of data streams and refines data representation for optimal learning.

2. RELATED WORK

2.1. Task-Free online continual learning

[21, 5] 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.

2.2. 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.

3. 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:

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

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).

4. METHODOLOGY

4.1. Feature-Distance based sample selection

Our proposed method is denoted as **FDBS** with M denoting the memory size and K the number of data samples so far streamed. 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:

- We begin by calculating the feature distance, denoted as $\boldsymbol{D}$ (refer to Eq. (4)), between every data point in the set $\mathbb{S}^{str}$ and each data sample stored in the memory set $\mathbb{S}^{mem}$. 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. (6)

$$\boldsymbol{D}_{i,j} = dist\left\{F(x_i), F(x_j)\right\}_{(x_i \in \mathbb{S}^{str}; x_j \in \mathbb{S}^{mem})} \quad (4)$$

- Subsequently, we compute $\boldsymbol{D}^{mem}$, as in Eq. (5), the feature distance between every pair of points in the memory set, and the minimum distance for each data point in the memory set in $\boldsymbol{d}^{mem}$, as shown in Eq. (6). We then calculate $\boldsymbol{a}$ as in Eq. (7) a weighted average distance 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.

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

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

- 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.

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

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.

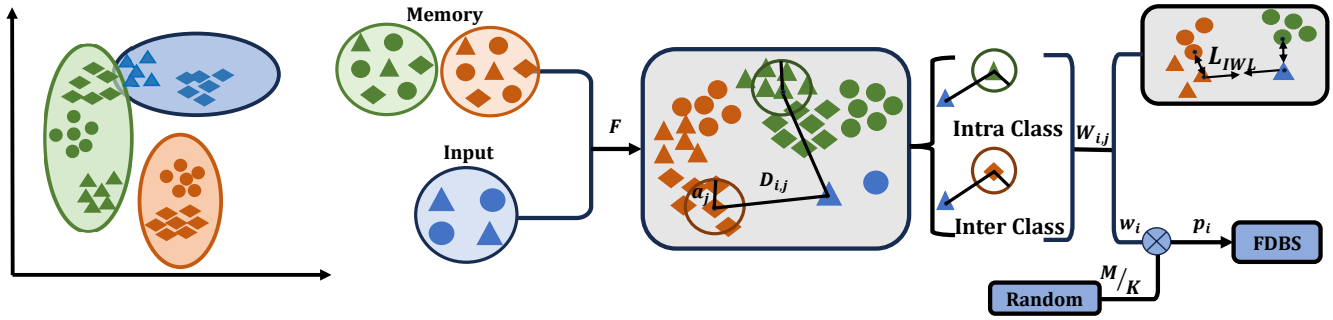
When comparing a new data sample x_i with a memory set data point x_j, we consider two scenarios based on their class labels. If they share the **same class label**, as shown in Fig. 1 (b), and if the feature distance $\boldsymbol{D}_{i,j}$ significantly exceeds $\boldsymbol{a}_j$, it implies a substantial 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.

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

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

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



(a) Inter-Class Similarity & Intra-Class Variance

(b) our **MSCL**

Fig. 1: Both figures use colors to represent domains and shapes for categories. Figure **(a)** shows models adapting to datasets with high inter-class similarity and intra-class variance, underscoring the challenge of differentiating closely related categories. Figure **(b)** introduces our proposed MSCL, which maps input data and a memory set into a shared feature space. Here, $\boldsymbol{D}_{i,j}$ denotes the distance between input data x_i and memory set data x_j. We calculate these distances to derive an importance weight matrix that quantifies the relative importance of each input data point in relation to those in the memory set, based on their intra-class diversity or inter-class similarity. These importance weights, combined with random selection, lead to our Feature-Distance based Sample Selection (FDBS). Using the importance weight matrix, we then develop a novel Contrastive Loss (IWL).

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.

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

4.2. Contrastive learning for better discriminative feature representation

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, we introduce a contrastive learning loss (IWL) to improve feature representation. 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. (11), optimizing $\boldsymbol{W}_{i,j}$ to align data points with matching class labels closer and separate those with differing labels.

$$L_{IWL} = \frac{\sum_{i=1}^{N_m} \sum_{j=1}^{N_b} log(\boldsymbol{W}_{i,j})}{\sum_{i=1}^{N_m} \sum_{j=1}^{N_b} \boldsymbol{\beta}_{i,j}} \tag{11}$$

In our algorithm, to reduce computational complexity, we do not fully update $\boldsymbol{D}^{mem}$ at each step. Instead, during each iteration, we draw a small batch of data from the memory set and dynamically update the corresponding distances and feature vectors for that specific batch.

5. EXPERIMENTS AND RESULTS

5.1. Balanced benchmarks

Building upon previous research [9, 7, 13], we utilize four well-established Continual Learning (CL) benchmarks: Split MNIST, Split ImageNet-1k, Split CIFAR-100, and PACS. Split MNIST comprises five tasks, each containing two classes. 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 ten 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.

5.2. Imbalanced benchmarks

Existing CL benchmarks, with uniform class and domain distributions, fail to test CL methods on non-stationary, imbal-

anced data. Thus, we've created benchmarks specifically to assess CL methods' robustness to data imbalance.

5.2.1. 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 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.

5.2.2. 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.

5.2.3. 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.

5.3. 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].

We compare Fine-tuning (FT), 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. Our method introduces Feature-Distance Based Sampling (FDBS) for choosing samples and Contrastive Learning Loss (IWL) for better representation learning. We test the effectiveness of both FDBS alone and combined with IWL in our experiments.

For MNIST, we utilize a two-hidden-layer MLP with 250 neurons per layer. Meanwhile, for all other datasets, we adopt the standard ResNet-18 architecture implemented in PyTorch. The replay buffer size is configured as 5000 for CIFAR-100, mini-ImageNet, and DomainNet, while it is set to 1000 for all other scenarios. We maintain a fixed batch size of 20 for the incoming data stream, with five update steps per batch. Notably, we abstain from employing data augmentation in our experiments. We set the σ value in our radial basis function (RBF) kernel at 0.5, and the τ value in Eq. (9) at 0.5.

5.4. Results on balanced benchmarks

Results for balanced scenarios are shown in Tab. 1. While the Experience Replay (ER) method fares well in these settings due to its unbiased memory selection, our proposed FDBS method paired with the Contrastive Learning Loss (IWL) offers notable improvements. This enhancement is largely attributed to IWL's feature space optimization, which aids FDBS's data sample selection based on feature space distance. The combination of FDBS and IWL also yields more consistent results, as evidenced by a reduced standard deviation. Especially for datasets like Rotated MNIST and PACS, FDBS excels by augmenting intra-class diversity in memory, thus increasing adaptability to domain shifts.

5.5. Results on imbalanced scenarios

Tab. 2 displays results in 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 FDBS 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. This advantage is amplified when paired with the IWL, reinforcing the benefits seen in balanced scenarios.

Table 1: 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. For Split CIFAR-100 and mini-ImageNet, the memory size was set to 5000, while for all other scenarios, the memory size was set to 1000.

Methods / Datasets	Split MNIST	mini ImageNet	Split CIFAR-100	PACS
Fine tuning	19.23 ± 0.32	4.21 ± 0.22	4.43 ± 0.17	20.56 ± 0.24
i.i.d. Offline	92.73 ± 0.21	52.52 ± 0.05	49.79 ± 0.28	56.94 ± 0.12
ER	81.68 ± 0.97	15.76 ± 2.34	18.26 ± 1.78	41.66 ± 1.45
GSS	80.38 ± 1.42	12.31 ± 1.26	13.57 ± 1.23	39.87 ± 3.25
CBRS	81.34 ± 1.27	15.58 ± 1.94	18.55 ± 1.68	41.34 ± 1.65
MIR	**86.76**± 0.67	16.73 ± 1.12	18.71 ± 0.89	42.2 ± 0.85
OCS	85.43 ± 0.86	16.59 ± 0.89	19.31 ± 0.48	42.63 ± 0.73
FDBS(ours)	85.79 ± 0.76	17.54 ± 2.17	19.89 ± 1.54	42.86 ± 1.37
MSCL(ours)	86.48 ± 0.57	**18.93** ± 0.74	**21.13** ± 0.94	**43.54** ± 0.75

Table 2: 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	3.18 ± 0.31	3.57 ± 0.25	15.54 ± 1.34	14.35 ± 1.23	2.35 ± 0.65
i.i.d. Offline	41.65 ± 0.57	43.17 ± 0.62	46.34 ± 0.47	46.18 ± 0.92	37.27 ± 0.73
ER	7.14 ± 0.81	8.25 ± 1.27	25.64 ± 2.19	22.48 ± 1.23	6.24 ± 0.62
GSS	8.38 ± 0.74	7.95 ± 0.48	24.46 ± 1.78	20.17 ± 2.14	5.15 ± 0.44
CBRS	10.21 ± 0.39	11.37 ± 0.63	25.97 ± 1.54	23.68 ± 1.75	6.13 ± 0.59
MIR	7.52 ± 0.93	8.97 ± 0.30	25.85 ± 2.19	22.15 ± 2.57	6.47 ± 0.45
OCS	11.68 ± 0.63	12.29 ± 0.49	27.15 ± 1.42	24.72 ± 1.37	8.47 ± 0.78
FDBS(ours)	12.35 ± 0.85	12.89 ± 0.62	29.13 ± 1.53	27.56 ± 1.52	10.25 ± 0.94
MSCL(ours)	**13.72**± 0.53	**14.21**± 0.34	**31.25**± 0.83	**28.64**± 1.44	**11.46**± 0.71

6. ABLATION STUDY

Our method includes two primary components: the memory selection method (FDBS) and the contrastive learning loss. In this section, we conduct a series of experiments on both balanced CIFAR-100 and imbalanced DomainNet to demonstrate the contributions and effectiveness of each component. The results of the ablation study are displayed in Tab. 3. We find that our proposed FDBS plays a more significant role in imbalanced scenarios, while both components collectively contribute to improved performance.

Table 3: Ablation studies on balanced CIFAR-100 and imbalanced DomainNet. We set the memory size to 5000.

Method	Balanced CIFAR-100	Imb DomainNet
Fine tuning	4.43 ± 0.17	2.35 ± 0.65
w/o L_{IWL}	19.89 ± 1.54	10.25 ± 0.94
w/o FDBS	20.25 ± 1.33	9.72 ± 0.86
MSCL	21.13 ± 0.94	11.46 ± 0.71

7. CONCLUSION

This paper introduces MSCL, a novel online CL approach combining Feature-Distance Based Sample Selection (FDBS) and Contrastive Learning Loss (IWL). FDBS picks examples by measuring distances between new and stored data, focusing on enhancing class diversity and boundary awareness. IWL aims to improve feature representation by adjusting intra-class and inter-class distances. Tests show MSCL outperforms existing memory-based CL methods in both balanced and imbalanced settings.

8. 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).

9. 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.