Faster convergence and Uncorrelated gradients in Self-Supervised Online Continual Learning

Koyo Imai[1], Naoto Hayashi[1], Tsubasa Hirakawa[1], Takayoshi Yamashita[1], and Hironobu Fujiyoshi[1]

Chubu University, 1200 Matsumoto-cho, Kasugai, Aichi, Japan
{kouyou,hayashi29,hirakawa}@mprg.cs.chubu.ac.jp,
{takayoshi,fujiyoshi}@isc.chubu.ac.jp

Abstract. Self-Supervised Online Continual Learning (SSOCL) focuses on continuously training neural networks from data streams. This presents a more realistic Self-Supervised Learning (SSL) problem setting, where the goal is to learn directly from real-world data streams. However, common SSL requires multiple offline training sessions with fixed IID datasets to acquire appropriate feature representations. In contrast, SSOCL involves learning from a non-IID data stream where the data distribution changes over time, and new data is added sequentially. Consequently, the challenges are insufficient learning with changing data distributions and the learning of inferior feature representations from non-IID data streams. In this study, we propose a method to address these challenges in SSOCL. The proposal method consists of a Multi-Crop Contrastive Loss, TCR Loss, and data selection based on cosine similarity to representative features. Multi-Crop Contrastive Loss and TCR Loss enable quick adaptation to changes in data distribution. Cosine similarity-based data selection ensures diverse data is stored in the replay buffer, facilitating learning from non-IID data streams. The proposed method shows superior accuracy compared to existing methods in evaluations using CIFAR-10, CIFAR-100, ImageNet-100, and CORe50.

Keywords: Online Continual Learning · Self-Supervised Learning · data stream

1 Introduction

In today's society, vast amounts of data are continually generated, such as data uploaded to the internet and frames continuously captured from in-vehicle cameras. Using this vast amount of data to train neural networks offers various advantages, including the accumulation of diverse knowledge and the reduction of costs associated with data collection. However, as shown in Fig. 1, such data streams have the characteristic that the data distribution changes over time and is non-Independent and Identically Distributed (IID). This contradicts the assumption that the data distribution a conventional neural network uses to train the dataset is stationary and IID. Online Continual Learning (OCL) aims to

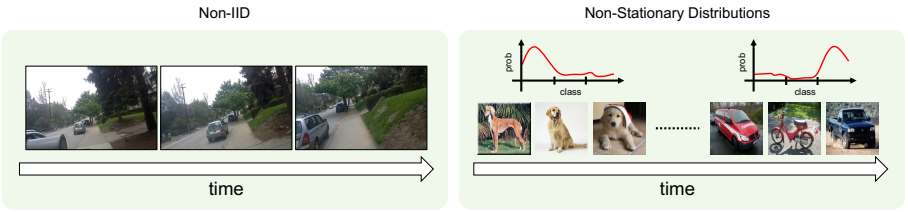
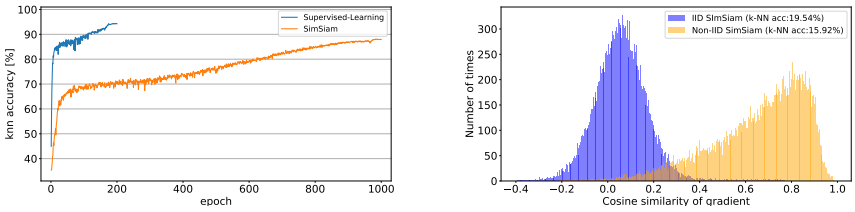


Fig. 1: Properties of real-world data streams include non-IID characteristics due to temporal correlation. For example, frames captured by an in-vehicle camera exhibit non-IID properties. Additionally, real-world data distributions are non-stationary and biased concerning the classes that appear over time.

address this problem and continuously learn models from data streams with the properties shown in Fig. 1. However, most methods use labeled data for training, and it is difficult to train directly from real-world data streams that have not been manually annotated.

Self-Supervised Online Continual Learning (SSOCL) involves learning from unlabeled data streams and aims to deploy these models in the real world. Self-Supervised Learning (SSL) is a type of unsupervised learning that can learn effective feature representations for downstream tasks from unlabeled data. Feature representations acquired by SSL have higher generalization performance than those acquired by supervised learning [26] and have attracted significant attention in continual learning [8,36,20], where new classes are added over time. However, SSOCL has problems as shown in Fig. 2a and Fig. 2b. As Fig. 2a shows, SSL converges more slowly than supervised learning, requiring more than $1,000$ epochs to reach 90% k-NN accuracy, while 200 epochs suffice for supervised learning. Therefore, SSL is difficult to learn sufficiently in an online environment where new data is generated sequentially and the data distribution changes over time. In addition, the data stream is non-IID, with data arriving successively, including extremely similar frames and a limited number of classes. Therefore, as Fig. 2b shows, in adjacent iterations, the similarity of the gradients during parameter updates is higher than when training on an offline IID dataset. As a result, models trained on non-IID data streams learn degraded feature representations with poor generalization performance compared with models trained on IID datasets.

In this paper, we propose a SSOCL method that addresses the slow convergence of SSL and can learn effectively on non-IID data streams. The proposed method consists of Multi-Crop Contrastive (MCC) loss, TCR Loss [35], and a selection of data to store in the replay buffer based on cosine similarity. Conventional SSL applies two types of data augmentations to a single image and calculates the loss based on the two resulting images [26,10,11,4,9,48,23]. On the other hand, MCC loss and TCR Loss apply three or more types of data augmentations to a single image and calculate the loss based on them. Increasing the number of crops speeds convergence and adapts quickly to changing data



(a) Learning Convergence Difference (b) Similarity of gradient

Fig. 2: Problems in Self-Supervised Online Continual Learning. (a)The k-NN accuracy during CIFAR-10 learning with SSL method SimSiam and supervised learning. The supervised learning is trained for 200 epochs, and SimSiam is trained for 1,000 epochs, with the k-NN accuracy measured for each epoch. (b)Similarity of the gradients during parameter updates in the t_{th} and $(t+1)_{th}$ iterations. CORe50 dataset (IID) and Seq-CORe50 (non-IID) are trained using the SSL method SimSiam.

distributions. Data selection using cosine similarity calculates the representative features of each data from the average of the features computed by the MCC loss. Then, it calculates cosine similarity for these representative features and removes redundant data with high similarity from the replay buffer. This ensures that diverse data with low similarity is retained in the replay buffer, preventing high similarity of gradients during parameter updates by learning from these diverse data. The proposed method improves the convergence speed of learning by increasing the number of crops[12,43], enabling the rapid learning of feature representations that are effective for data selection and the retention of a greater variety of data in the replay buffer.

The contributions of this paper are as follows.

1. We propose a new Self-Supervised Online Continual Learning method that can address two problems in Self-Supervised Online Continual Learning: convergence speed and gradient similarity during parameter updates.
2. The proposed method improves the convergence speed of Self-Supervised Learning, enabling quick adaptation to changes in data distribution in online learning with unlabeled data streams.
3. The proposal method can select diverse data from the data stream and store them in a replay buffer, and uncorrelate gradients during parameter updates. This prevents the learning of degraded representations in non-IID data streams.
4. Comprehensive experiments using CIFAR-10, CIFAR-100, ImageNet-100, and CORe50 demonstrate that our method outperforms conventional methods.

2 Related Work

In this section, we systematically summarize Self-Supervised Learning and continual learning, which are closely related to this study.

2.1 Self-Supervised Learning

Self-Supervised Learning (SSL) is a method of offline learning from unannotated data, which pretrains feature representations useful for various downstream tasks. The feature representations acquired through SSL exhibit higher generalization performance compared to those acquired through supervised learning, achieving performance equal to or exceeding that of supervised learning in downstream tasks [26,10,11,4,9,17,6,48,7]. Contrastive learning[9,10,26] applies different data augmentations to create positive pairs that potentially have the same label. The model learns to ensure that the feature representations of these positive pairs are similar, while distinguishing them from those of other data (negative pairs). A negative-free method that does not use negative examples has also been proposed [11,23,7]. Other methods include those that reduce redundancy in feature representation [48,18,4] and those based on clustering [6]. However, these SSL methods typically require hundreds of epochs of training on a fixed dataset in an offline environment to achieve accuracy comparable to supervised learning on downstream tasks. They do not focus on learning in an online setting.

2.2 Continual Learning

Continual learning [40,16,3,39,20,5,42] involves using sequentially acquired datasets containing new task data while discarding previously learned datasets and using the data in the new datasets for training. Sequentially arriving datasets contain data for different tasks, and the goal of continual learning is to acquire knowledge of new tasks while retaining knowledge of previous tasks. Methods of continual learning can be broadly classified as replay-based, regularization-based, and architecture-based. Replay-based methods use a replay buffer [3,40,28,5] or generative model [42,22,13] to learn from both past task data and new task data simultaneously, thereby mitigating catastrophic forgetting. Regularization-based methods prevent the disappearance of previously acquired knowledge by constraining changes in the model's output[5,40,16,31] or by restricting the variation of parameters[32,29]. Architecture-based methods use different parameters for different tasks[41,19].

In contrast to normal continual learning, which trains on datasets for multiple epochs in an offline environment, Online Continual Learning (OCL) [14,49,45,44,1] learns from a stream of data in an online environment using a single pass. Data arrive sequentially in small batches, and a batch can only be trained for a limited amount of time before the next batch arrives. Old batches are discarded after the arrival of a new batch, and it is not possible to reuse discarded batches for further learning. Similar to common continual learning, many replay-based

and regularization-based methods have been proposed [24,25,49,14]. OCL aims to learn directly from real-world data streams. However, many traditional OCL methods focus on labeled data streams, making it challenging to learn from unlabeled data streams in the real world.

Self-Supervised Online Continual Learning (SSOCL) continuously learns using unlabeled data streams [47,38,2,46]. This study aims to learn directly from real-world data streams with the characteristics shown in Fig. 1. SCALE [47] mitigates catastrophic forgetting [21] in changing data distributions without using labels or prior knowledge by using Pseudo-Supervised Contrastive Loss and Self-Supervised Forgetting Loss. However, it does not address the slow convergence of SSL. MinRed [38] demonstrated the ability to learn from non-IID data streams by using cosine similarity for data selection and maintaining diverse data in the replay buffer. However, it does not address the convergence speed of SSL, and since it uses pre-trained models when learning from data streams with changing distributions, insufficient evidence exists regarding its ability to learn directly from the data streams. RALS [2] addresses the slow convergence of SSL in data streams by dynamically adjusting hyperparameters. This approach improves accuracy in single-pass learning with data streams, but it does not explicitly address non-IID data streams.

Existing SSOCL methods address either the challenge of learning with non-IID data distributions or the slow convergence of SSL, but no method addresses both problems simultaneously. To the best of our knowledge, the method proposed in this paper is the first SSOCL approach that explicitly addresses both problems. Note that this paper focuses on single-pass learning using the limited data available in the data stream and does not address the computational processing required for learning.

3 Problem Formulation

In this section, we formulate the problem of Self-Supervised Online Continual Learning (SSOCL). In this paper's problem setting, the objective is to learn a model f_θ with parameters θ from real-world data streams. Therefore, no prior knowledge such as labels and task IDs is used.

The training data is incrementally provided from the data stream and the training batch at a given time t is denoted as X^t. The training batch X^t contains data for a batch size b, where $X^t = [x^1, x^2, ..., x^b]$. Data contained in X^t will not be included in batches other than X^t. Learning with X^t is only possible until the training batch X^{t+1} arrives at the next time $t+1$, after which X^t is discarded. Let K be the number of times the model can learn from the data X^t before it is discarded. The model trains K times on X^t. Discarded data cannot be used for subsequent training and must be trained in a single pass.

Data streams in the real-world can be broadly classified into two types: (i) Data collected in the wild or the vast amount of data uploaded daily to the internet exhibit changing data distributions over time, with weak correlations among the data within a batch X^t. (ii) Continuous frames captured by

cameras mounted on agents such as cars or humans, which have strong correlations among the data within batch X^t. In this paper, the first type of data stream is replicated using CIFAR-10, CIFAR-100, and ImageNet-100, while the second type of data stream is replicated using CORe50. When the data distribution changes, the model begins learning on the new data distribution without sufficient training on the previous one. Additionally, when there is a strong correlation among the data within a batch, the model's parameters may become overly fitted to specific data. The learning objective in this problem setting is to quickly adapt to changes in data distribution and to learn effective feature representations from strongly correlated and redundant data streams.

4 Proposal Method

We propose a method to address two issues in Self-Supervised Online Continual Learning (SSOCL): the slow convergence of Self-Supervised Learning (SSL) and the correlation of gradients during parameter updates from data streams. The method learns online from unlabeled data streams. The proposed method consists of two main components: two types of self-supervised losses, Total Coding Rate (TCR) [35] and Multi-Crop Contrastive (MCC) loss, and the selection of data to be stored in the replay buffer based on cosine similarity. MCC loss and TCR loss address the slow convergence of SSL, as shown in Fig. 2a, and help learn better feature representations in single-pass data streams. Additionally, a replay buffer utilizing cosine similarity for data selection is used to choose diverse data from the data stream and retain it in the replay buffer. To address the challenge shown in Fig. 2b, the proposal method learns only from the diverse data stored in the replay buffer. This alleviates the bias in parameter updates caused by learning from non-IID data streams.

MCC Loss and TCR Loss enable rapid learning of better feature representations, which are then used to select the data to be stored in the replay buffer. By accelerating learning convergence, it is possible to learn meaningful feature representations even in data streams where the data distribution changes over time. Furthermore, by using cosine similarity to select data based on the learned meaningful feature representations, selecting and retaining diverse data is possible from single-pass data streams. The diverse data retained in the replay buffer prevents learning from biased data in temporal data streams, leading to further improvements in feature representations. Thus, the proposed method improves performance through the cooperation of its two components.

Fig. 3 shows the pipeline of the proposed method in detail. The size of the replay buffer is M. When the number of retained data exceeds M, data selection using cosine similarity is performed to choose the data to keep in the replay buffer, and the rest are discarded. A batch of data $X = [x^1, x^2, \ldots, x^b]$ is extracted from the replay buffer to form a mini-batch for training. $X_1, X_2, \ldots, X_N$ are generated by applying N types of data augmentation to mini-batch X, where $X_i = [x_i^1, x_i^2, \ldots, x_i^b]$. Each augmented mini-batch X_i is input into the model f_θ to obtain the features $Z_i = f_\theta(X_i)$, where $Z_i = [z_i^1, z_i^2, \ldots, z_i^b]$ and

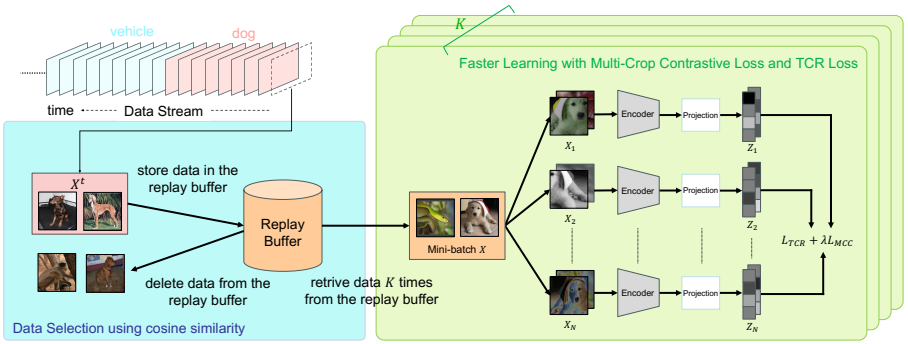


Fig. 3: The pipeline of proposal method. The blue part is for data selection, and the green part is for faster learning convergence.

$Z = [Z_1, Z_2, ..., Z_N]$. Using the output features Z, MCC and TCR loss are calculated, and the model parameters θ are updated. The process of extracting data X from the replay buffer and updating the model parameters θ is repeated K times until new data input is received from the data streams. In the following, we provide a detailed explanation of each component in the proposed method.

4.1 Multi-Crop Contrastive Loss and Total Coding Rate Loss

The proposed method uses Multi-Crop Contrastive (MCC) and TCR loss as its loss function, which is formulated by:

$$\mathcal{L} = \mathcal{L}_{TCR} + \lambda \mathcal{L}_{MCC} \tag{1}$$

where λ balances the two losses. $\mathcal{L}_{MCC}$ and $\mathcal{L}_{TCR}$ do not require explicit labels and can learn from unlabeled data streams.

Multi-Crop Contrastive Loss. MCC loss is a contrastive loss based on 2-crop ($N = 2$) losses like InfoNCE Loss [37] and NT-Xent Loss [9], but it can be calculated using multiple crops ($N > 3$). The positive pairs $\left(x_1^j, x_2^j\right)$ loss in NT-Xent Loss is defined as:

$$\mathcal{L}^j = -\log\left(\frac{\exp(\mathrm{sim}(\mathrm{z}_1^{\mathrm{j}} \cdot \mathrm{z}_2^{\mathrm{j}})/\tau)}{\sum_{k=1,k\neq j}^{b}\exp(z_1^j \cdot z_1^k/\tau) + \sum_{k=1}^{b}\exp(z_1^j \cdot z_2^k/\tau)} + \frac{\exp(\mathrm{sim}(\mathrm{z}_2^{\mathrm{j}} \cdot \mathrm{z}_1^{\mathrm{j}})/\tau)}{\sum_{k=1}^{b}\exp(z_2^j \cdot z_1^k/\tau) + \sum_{k=1,k\neq j}^{b}\exp(z_2^j \cdot z_2^k/\tau)}\right) \tag{2}$$

where z is a feature, sim is a pairwise similarity and τ is a temperature parameter. In traditional contrastive loss, two different data augmentations are applied to a single data to create positive pairs. The model learns to bring the feature representations of the positive pairs closer together, while pushing those of the negative pairs, which are other data, further apart. The traditional contrastive

loss, shown in Eq. (2), requires multiple iterations of learning the same data until convergence, as it brings the feature representations of positive pairs closer together in a one-to-one manner. Fast-MoCo[12] and EMP-SSL[43] improve the speed of convergence in SSL by increasing the number of positive feature representations that are brought closer together at one time. MCC loss is based on the idea of improving the speed of convergence by increasing the number of data feature representations that are brought closer together at one time and is defined as follows:

$$\mathcal{L}_{MCC} = \frac{1}{Nb} \sum_{i=1}^{N} \sum_{j=1}^{b} \left(-\log \frac{\exp(\bar{z}^j \cdot z_i^j / \tau)}{\sum_{k=1}^{N} \sum_{l=1}^{b} \exp(\bar{z}^j \cdot z_k^l / \tau)} \right) \tag{3}$$

where $\bar{z}^j$ is the average of all feature representations of x^j with data augmentations, calculated as follows:

$$\bar{z}^j = \frac{1}{N} \sum_{i=1}^{N} z_i^j. \tag{4}$$

MCC loss brings the feature representations of the data points $x_1^j, x_2^j, \ldots, x_N^j$, which are created by applying data augmentations to the data x^j, closer to their averages $\bar{z}^j$ instead of bringing them closer individually. Additionally, the model learns to push away the feature representations z_k^l, which are created by applying data augmentation to different data, from the average feature representations $\bar{z}^j$.

Total Coding Rate Loss. Total Coding Rate (TCR) loss [35] regularizes the covariance of the feature representations, uncorrelated each dimension of the learned features. VICReg [4] enhances the similarity of positive feature representations while regularizing the covariance to obtain better feature representations. The proposed method is based on the VICReg concept and uses TCR loss to regularize the covariance of the feature representations, along with MCC loss to directly bring feature representations closer or push them apart. This enables the acquisition of better feature representations in single-pass learning from data streams. TCR loss is defined as follows:

$$\mathcal{L}_{TCR} = \frac{1}{N} \sum_{i=1}^{N} \left(-\frac{1}{2} \log \det \left(I + \frac{d}{b\epsilon^2} Z_i Z_i^T \right) \right) \tag{5}$$

where I is an identity matrix, d is the number of output dimensions of the Projection, and ϵ is a hyperparameter representing distortion.

4.2 Data selection using cosine similarity for representative features

Data selection using cosine similarity selects diverse data from the data stream and stores it in the replay buffer. As shown in Fig. 1, data streams are non-IID, and learning from them directly results in continuously learning similar data. As a result, the gradients during parameter updates become highly similar. This

leads to the model parameters becoming overly fitted to specific data distributions, resulting in learning feature representations with low generalization performance that do not work well on other data distributions. To prevent such issues, diverse data is retained in the replay buffer, and learning from this diverse data helps prevent overfitting to a specific data distribution. The simplest way to retain diverse data is to select and store an equal number of data from each class [14]. However, in real-world data streams without labels, such label-based data selection is impossible. When selecting data from unlabeled data streams, clustering the feature representations or calculating the distance between data using metrics such as cosine similarity or Chebyshev distance can be used to select the data to [47,38,33]. In the proposed method, data selection for the replay buffer is performed using cosine similarity on the feature representations of the data.

In the proposed method, the replay buffer stores not only the data input from the data stream but also the representative feature representations of that data. OnPro[44] uses the average of the feature representations of all data in class i as the representative feature for that class. Additionally, CoPE [14] adapts to changes in representative features caused by changes in model parameters and data distribution in the data stream by taking the moving average of the mean of new features and the mean of old features. In the proposed method, the representative feature of data x_i is calculated using the average of the feature representations computed by Eq. (4). The representative feature of the data retained in the replay buffer is defined as follows:

$$\bar{z}_i^* = \alpha \bar{z}_i^* + (1 - \alpha)\, \bar{z}_i \tag{6}$$

where $\bar{z}_i^*$ is the representative feature of data x_i retained in the replay buffer, α is the moving average coefficient, and $\bar{z}_i$ is the average feature representation of the data x_i calculated by Eq. (4). Using the average feature representation calculated when computing the MCC loss in Eq. (3) can reduce the computational cost. Data selection using the representative feature calculated by Eq. (6) is formulated as follows:

$$x_i^* = \arg \min_{x_i \in \mathcal{M}} \min_{x_j \in \mathcal{M}} \mathrm{Sim}\left(\bar{z}_i^*, \bar{z}_j^*\right) \tag{7}$$

where $\mathcal{M}$ is the replay buffer, x_i^* is the data to be stored in the replay buffer, and $\mathrm{Sim}(,)$ is the cosine similarity. Cosine similarity is calculated for the representative features of the data in the buffer, and by retaining the data with low similarity in the replay buffer, an IID dataset is constructed within it. Thus, learning from the data in the buffer enables learning even in non-IID data streams.

5 Experiment

5.1 Experimental Setup

Datasets. The datasets used are CIFAR-10, CIFAR-100 [30], and ImageNet-100 [15]. Each dataset is constructed as sequential (Seq), sequential blurred (Seq-bl), and sequential imbalance (Seq-im) data streams following [47]. Additionally,

CORe50 [34] is selected as a non-IID data stream and used as the Seq-CORe50 data stream.

Networks. ResNet-18[27] with an output dimension of 512 is used for all datasets. The proposal method connects a projector with a hidden layer dimension of 4,096 and an output layer of 1,024 behind ResNet-18.

Baselines. The Self-Supervised Online Continual Learning (SSOCL) methods MinRed [38] and SCALE [47], as well as the Self-Supervised Learning (SSL) method EMP-SSL [43] that can be trained in one epoch, are used as baselines. EMP-SSL uses the same replay buffer as the proposed method to verify the effectiveness of the Multi-Crop Contrastive (MCC) loss. SSOCL and normal Self-Supervised Continual Learning (SSCL) methods are not used as baselines because their effectiveness cannot be fully demonstrated due to differences in problem settings. Optimization methods and data augmentations follow the settings in each respective paper.

Evaluation Metric. k-NN classifier, which does not require retraining, is used as the evaluation metric, and accuracy is evaluated on the validation dataset.

Hyperparameters. Unless otherwise noted, the hyperparameters used in the experiments meet the following settings. The batch size arriving from the data stream is set to 100. For all methods, the replay buffer size is set to 1,024, and the mini-batch size extracted from the replay buffer is set to 100. The number of iterations K for learning before a new batch arrives from the data stream is set to 3 for the proposed method and EMP-SSL, 20 for SCALE, and 40 for MinRed. The hyperparameter λ used in the loss function of the proposed method is set to 200, and the number of crops N is set to 20, and α is set to 0.5. Other implementation details are provided in the Supplementary Materials.

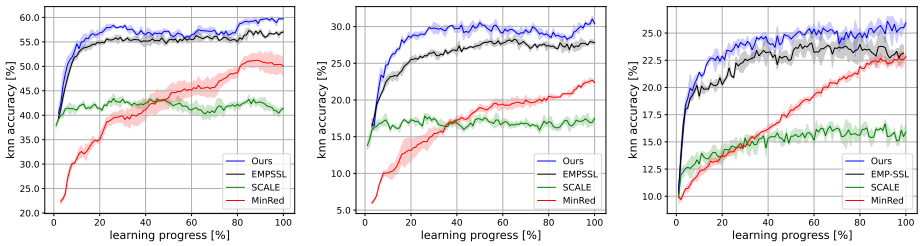
5.2 CIFAR-10, CIFAR-100, and ImageNet-100

Final accuracy. Table 1 shows the k-NN accuracy at the end of training for the models trained on the CIFAR-10, CIFAR-100, and ImageNet-100 data streams. Training is performed three times with different seed values, and the accuracy represents the average of these three runs. Table 1 shows that the proposed method achieved higher accuracy than the conventional methods across all streams. The proposed method improved accuracy by up to 18.3pt, 19.11pt, and 17.36pt for the different settings of CIFAR-10. For CIFAR-100, it achieved maximum improvements of 12.92pt, 13.39pt, and 12.97pt. For ImageNet-100, it achieved maximum improvements of 10.35pt, 10.23pt, and 9.47pt. MinRed has a lower accuracy on the Seq-im data stream compared with the Seq and Seq-bl data streams. This result likely stems from varying numbers of data available for training across different data distributions, leading to insufficient learning for data distributions with fewer data. In contrast, the proposed method maintains consistent accuracy across different data streams, indicating its capability to learn regardless of the type of data stream.

Comparison in the learning process. To investigate the speed of learning convergence, we compare the k-NN accuracy during the training process for each dataset. Fig. 4 shows the k-NN accuracy transitions during the training

Table 1: k-NN accuracy [%] at the end of training for each data stream

	CIFAR-10			CIFAR-100			ImageNet-100		
	Seq	Seq-bl	Seq-im	Seq	Seq-bl	Seq-im	Seq	Seq-bl	Seq-im
MinRed[38]	50.04	51.18	46.41	22.38	23.20	21.26	22.87	22.71	20.46
SCALE[47]	41.41	40.85	41.31	17.49	16.93	17.03	15.46	15.41	15.77
EMP-SSL[43]	57.02	57.32	57.31	27.81	28.40	27.90	22.79	22.01	22.99
Ours	**59.71**	**59.96**	**58.67**	**30.41**	**30.32**	**30.00**	**25.81**	**25.64**	**25.24**



(a) Seq-CIFAR10 (b) Seq-CIFAR100 (c) Seq-ImageNet100

Fig. 4: k-NN accuracy in the learning process. The vertical and horizontal axes represent k-NN accuracy and training progress, respectively. The solid line represents the average accuracy of three runs with different seed values, and the shaded area represents its standard deviation.

process for Seq-CIFAR-10, Seq-CIFAR-100, and Seq-ImageNet-100 data streams. Fig. 4 shows that the k-NN accuracy of the proposed method at 20% progress in training is approximately 56.00% for Seq-CIFAR-10, 28.00% for Seq-CIFAR-100, and 22.50% for Seq-ImageNet-100. This is equal to or better than the accuracy of other methods at the end of their training. Compared with EMP-SSL, which can be trained in one epoch, the proposed method achieves a higher accuracy during the training process. Furthermore, it demonstrates improved accuracy as training progresses, indicating an improved convergence speed and a high adaptability to changes in data distribution.

5.3 CORe50

We will compare the gradient similarity of parameter updates and the k-NN accuracy during training on the Seq-CORe50 data stream. The replay buffer sizes are set to 1,024, 2,048, and 4,096, and the number of rehearsal iterations K is set to 5, 10, and 20.

Final accuracy. Table 2 shows the k-NN accuracy at the end of training for each method, in which the proposed method achieves the highest accuracy for each value of K. These results show that the proposed method is more effective than

Table 2: k-NN accuracy [%] at the end of training for Seq-CORe50

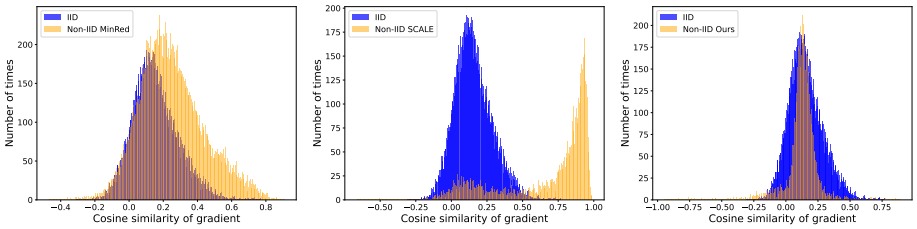
K	Method	Buffer Size 1,024	2,048	4,096
5	MinRed[38]	16.52	15.87	16.31
	SCALE[47]	14.99	15.48	16.02
	Ours	**18.06**	**22.05**	**22.68**
10	MinRed[38]	17.25	16.61	17.46
	SCALE[47]	16.88	16.35	16.57
	Ours	**19.73**	**22.06**	**23.55**
20	MinRed[38]	18.93	17.39	18.71
	SCALE[47]	15.30	15.23	15.20
	Ours	**20.92**	**23.16**	**23.47**

conventional methods for learning on non-IID data streams. Focusing on $K = 5$, the accuracy of conventional methods increases slightly or remains constant as the buffer size increases from 1,024 to 4,096. In contrast, the proposed method shows an accuracy increase of 4.62pt with the increase in buffer size, indicating the highest benefit from the buffer size increase.

Similarity of gradient during parameter updates. Fig. 5 shows the gradient similarity histograms during parameter updates of each method. Fig. 5 shows that SCALE has a higher gradient similarity during Seq-CORe50 training. Also, Table 2 shows that the accuracy of SCALE decreases when the value of K is increased from 10 to 20. These observations indicate that increasing K causes SCALE to train on biased data, resulting in decreased accuracy. On the other hand, increasing the value of K improves accuracy for each buffer size in both the proposed method and MinRed, with MinRed showing an improvement of up to 2.41pt and the proposed method showing an improvement of up to 2.86pt. Fig. 5 shows that MinRed has a higher gradient similarity when trained with Seq-CORe50 than when trained with IID-CORe50. In contrast, the proposed method does not exhibit higher gradient similarity when trained on Seq-CORe50, and the histogram is similar to that obtained when trained on IID-CORe50. These results show that the proposed method is able to address gradient bias, which is a problem when training on non-IID data streams.

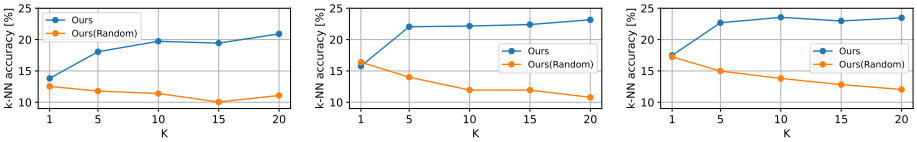
5.4 Ablation Study

We investigate the effectiveness of the data selection method used in the replay buffer of the proposed method. We compare the proposed method and the proposed method with random data selection. The dataset used is Seq-CORe50, and the accuracy is compared while changing the buffer size. The comparison results are shown in Fig. 6. When random data selection is used, increasing the value of K results in a decrease in accuracy regardless of buffer size. This is because biased data was stored in the replay buffer and used for learning. In contrast, the proposed method often shows improved accuracy as the value of K



(a) MinRed (b) SCALE (c) Ours

Fig. 5: Gradient similarity between the t_{th} and $(t+1)_{th}$ iterations during the training of the Seq-CORe50 data stream. The gradient similarity when learning the Seq-CORe50 data stream with each method, along with that when training SimSiam on the IID-CORe50, are displayed simultaneously.



(a) Buffer Size : 1,024 (b) Buffer Size : 2,048 (c) Buffer Size : 4,096

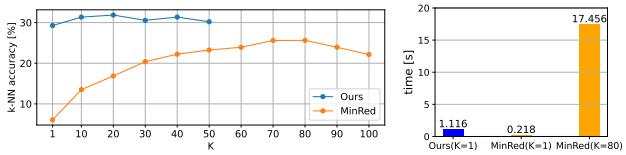
Fig. 6: k-NN Accuracy with different Data Selection Methods.

is increased. This is because more diverse data were kept in the buffer and were used for training. These results confirm the effectiveness of the data selection method of the proposed method.

5.5 Analysis

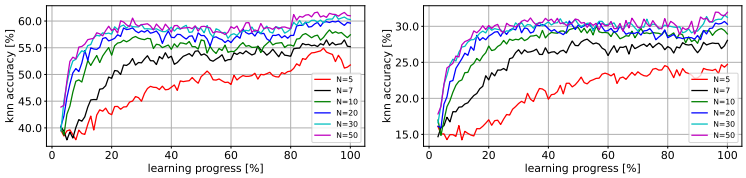
The number of rehearsal iterations K and the number of crops N are important in the proposed method and SSOCL methods. We investigate the impact of K and N on SSOCL and confirm the effectiveness of the proposed method.

The result of the experiment with changes in the number of rehearsal iterations K are shown in Fig. 7. Fig. 7a shows that increasing the value of K generally leads to improved accuracy. It is also observed that when the value of K exceeds a certain point, the accuracy remains constant or decreases. This is because increasing the number of training sessions with limited data leads to a loss of the model's generalization performance. This result shows that there is a limit to the improvement of accuracy by increasing K. On the other hand, comparing the proposed method at $K = 1$ with MinRed's highest accuracy at $K = 80$, we can confirm that the proposed method is about 4.0 pt higher. Furthermore, Fig. 7b shows that the learning time of MinRed when $K = 80$ is about 15 times longer than that of the proposed method when $K = 1$. Considering the problem setting of SSOCL, the proposed method is more suitable for SSOCL because it can be learned in a small number of times and in short learning time.



(a) k-NN accuracy (b) training time

Fig. 7: k-NN accuracy and training time for Seq-CIFAR-100 with changing the number of rehearsal iterations K. (a) shows the k-NN accuracy for different values of K. The value of K varies from 1 to 50 for the proposed method and from 1 to 100 for MinRed. (b) shows the time it takes to train before new data arrives from the data stream.



(a) Seq-CIFAR-10 (b) Seq-CIFAR-100

Fig. 8: k-NN accuracy when changing the number of crops N. The vertical axis represents k-NN accuracy, and the horizontal axis represents training progress.

The results of the experiment with changes in the number of crops N are shown in Fig. 8. Fig. 8 shows that the accuracy of the proposed method improves as the number of crops N is increased. These results show that the proposed method can improve accuracy in learning with only limited data from a data stream, a challenge for existing methods.

6 Conclusion

This paper demonstrates that Self-Supervised Online Continual Learning from real-world data streams faces challenges such as adapting to change distributions and learning degraded feature representations from non-IID data streams. The proposed method addresses these issues through a Multi-Crop Contrastive Loss, TCR Loss, and data selection based on cosine similarity to representative features. The proposed method demonstrated significant accuracy improvements compared to conventional methods through evaluation experiments using CIFAR-10, CIFAR-100, ImageNet-100, and CORe50. Also, the proposed method can potentially enhance the effectiveness of conventional methods using knowledge distillation by addressing the problem of insufficient learning in OCL.

References

1. Aljundi, R., Lin, M., Goujaud, B., Bengio, Y.: Gradient based sample selection for online continual learning. In: Advances in Neural Information Processing Systems. Curran Associates, Inc. (2019)
2. Azar, S.M., Timofte, R.: Speeding up online self-supervised learning by exploiting its limitations. In: Köthe, U., Rother, C. (eds.) Pattern Recognition. pp. 476–490. Springer Nature Switzerland, Cham (2024)
3. Bang, J., Kim, H., Yoo, Y., Ha, J.W., Choi, J.: Rainbow memory: Continual learning with a memory of diverse samples. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8218–8227 (June 2021)
4. Bardes, A., Ponce, J., LeCun, Y.: VICReg: Variance-invariance-covariance regularization for self-supervised learning. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=xm6YD62D1Ub
5. Buzzega, P., Boschini, M., Porrello, A., Abati, D., CALDERARA, S.: Dark experience for general continual learning: a strong, simple baseline. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 15920–15930. Curran Associates, Inc. (2020)
6. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 9912–9924. Curran Associates, Inc. (2020)
7. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9650–9660 (October 2021)
8. Cha, H., Lee, J., Shin, J.: Co2l: Contrastive continual learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9516–9525 (October 2021)
9. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 1597–1607. PMLR (13–18 Jul 2020), https://proceedings.mlr.press/v119/chen20j.html
10. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020)
11. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15750–15758 (June 2021)
12. Ci, Y., Lin, C., Bai, L., Ouyang, W.: Fast-moco: Boost momentum-based contrastive learning with combinatorial patches. In: European Conference on Computer Vision. pp. 290–306. Springer (2022)
13. Cong, Y., Zhao, M., Li, J., Wang, S., Carin, L.: Gan memory with no forgetting. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 16481–16494. Curran Associates, Inc. (2020)
14. De Lange, M., Tuytelaars, T.: Continual prototype evolution: Learning online from non-stationary data streams. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 8250–8259 (October 2021)

15. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
16. Dhar, P., Singh, R.V., Peng, K.C., Wu, Z., Chellappa, R.: Learning without memorizing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
17. Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., Zisserman, A.: With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9588–9597 (October 2021)
18. Ermolov, A., Siarohin, A., Sangineto, E., Sebe, N.: Whitening for self-supervised representation learning. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 3015–3024. PMLR (18–24 Jul 2021), https://proceedings.mlr.press/v139/ermolov21a.html
19. Fernando, C., Banarse, D., Blundell, C., Zwols, Y., Ha, D., Rusu, A.A., Pritzel, A., Wierstra, D.: Pathnet: Evolution channels gradient descent in super neural networks. arXiv preprint arXiv:1701.08734 (2017)
20. Fini, E., da Costa, V.G.T., Alameda-Pineda, X., Ricci, E., Alahari, K., Mairal, J.: Self-supervised models are continual learners. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9621–9630 (June 2022)
21. French, R.M.: Catastrophic forgetting in connectionist networks. Trends in Cognitive Sciences **3**(4), 128–135 (1999). https://doi.org/https://doi.org/10.1016/S1364-6613(99)01294-2
22. Gao, R., Liu, W.: Ddgr: Continual learning with deep diffusion-based generative replay. In: International Conference on Machine Learning (2023), https://api.semanticscholar.org/CorpusID:260816086
23. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Advances in neural information processing systems **33**, 21271–21284 (2020)
24. Gu, Y., Yang, X., Wei, K., Deng, C.: Not just selection, but exploration: Online class-incremental continual learning via dual view consistency. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7432–7441 (2022). https://doi.org/10.1109/CVPR52688.2022.00729
25. He, J., Mao, R., Shao, Z., Zhu, F.: Incremental learning in online scenario. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
26. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
27. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
28. Hou, S., Pan, X., Loy, C.C., Wang, Z., Lin, D.: Learning a unified classifier incrementally via rebalancing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)

29. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences **114**(13), 3521–3526 (2017)
30. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
31. Li, Z., Hoiem, D.: Learning without forgetting. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(12), 2935–2947 (2018). https://doi.org/10.1109/TPAMI.2017.2773081
32. Lin, G., Chu, H., Lai, H.: Towards better plasticity-stability trade-off in incremental learning: A simple linear connector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 89–98 (June 2022)
33. Lin, Z., Wang, Y., Lin, H.: Continual contrastive learning for image classification. In: 2022 IEEE International Conference on Multimedia and Expo (ICME). pp. 1–6 (2022). https://doi.org/10.1109/ICME52920.2022.9859995
34. Lomonaco, V., Maltoni, D.: Core50: a new dataset and benchmark for continuous object recognition. In: Levine, S., Vanhoucke, V., Goldberg, K. (eds.) Proceedings of the 1st Annual Conference on Robot Learning. Proceedings of Machine Learning Research, vol. 78, pp. 17–26. PMLR (13–15 Nov 2017), https://proceedings.mlr.press/v78/lomonaco17a.html
35. Ma, Y., Derksen, H., Hong, W., Wright, J.: Segmentation of multivariate mixed data via lossy data coding and compression. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(9), 1546–1562 (2007). https://doi.org/10.1109/TPAMI.2007.1085
36. Madaan, D., Yoon, J., Li, Y., Liu, Y., Hwang, S.J.: Representational continuity for unsupervised continual learning. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=9Hrka5PA7LW
37. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
38. Purushwalkam, S., Morgado, P., Gupta, A.: The challenges of continuous self-supervised learning. In: European Conference on Computer Vision. pp. 702–721. Springer (2022)
39. Rajasegaran, J., Khan, S., Hayat, M., Khan, F.S., Shah, M.: itaml: An incremental task-agnostic meta-learning approach. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
40. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
41. Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. arXiv preprint arXiv:1606.04671 (2016)
42. Shin, H., Lee, J.K., Kim, J., Kim, J.: Continual learning with deep generative replay. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)
43. Tong, S., Chen, Y., Ma, Y., Lecun, Y.: Emp-ssl: Towards self-supervised learning in one training epoch. arXiv preprint arXiv:2304.03977 (2023)
44. Wei, Y., Ye, J., Huang, Z., Zhang, J., Shan, H.: Online prototype learning for online continual learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 18764–18774 (October 2023)

45. Yan, H., Wang, L., Ma, K., Zhong, Y.: Orchestrate latent expertise: Advancing online continual learning with multi-level supervision and reverse self-distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 23670–23680 (June 2024)
46. Yu, X., Rosing, T., Guo, Y.: Evolve: Enhancing unsupervised continual learning with multiple experts. In: 2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 2355–2366. IEEE Computer Society, Los Alamitos, CA, USA (jan 2024). https://doi.org/10.1109/WACV57701.2024.00236, https://doi.ieeecomputersociety.org/10.1109/WACV57701.2024.00236
47. Yu, X., Guo, Y., Gao, S., Rosing, T.: Scale: Online self-supervised lifelong learning without prior knowledge. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 2484–2495 (June 2023)
48. Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow twins: Self-supervised learning via redundancy reduction. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 12310–12320. PMLR (18–24 Jul 2021)
49. Zhang, Y., Pfahringer, B., Frank, E., Bifet, A., Lim, N.J.S., Jia, Y.: A simple but strong baseline for online continual learning: Repeated augmented rehearsal. In: Advances in Neural Information Processing Systems. vol. 35, pp. 14771–14783. Curran Associates, Inc. (2022)