REMEMORY-BASED SIMSIAM FOR UNSUPERVISED CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised continual learning (UCL) has started to draw attention from the continual learning community, motivated by the practical need of representation learning with unlabeled data on sequential tasks. However, most of recent UCL methods focus on mitigating the catastrophic forgetting problem with a replay buffer to store previous data (i.e., rehearsal-based strategy), which needs much extra storage and thus limits their practical applications. To overcome this drawback, based on contrastive learning via SimSiam, we propose a novel rememory-based SimSiam (RM-SimSiam) method to reduce the dependency on replay buffer under the UCL setting. The core idea of our RM-SimSiam is to store and remember the old knowledge with a data-free historical module instead of replay buffer. Specifically, this historical module is designed to store the historical average model of all previous models (i.e., the memory process) and then transfer the knowledge of the historical average model to the new model (i.e., the rememory process). To further improve the rememory ability of our RM-SimSiam, we devise an enhanced SimSiam-based contrastive loss by aligning the representations outputted by the historical and new models. Extensive experiments on three benchmarks demonstrate the effectiveness of our RM-SimSiam under the UCL setting.

1 INTRODUCTION

Continual learning (Ring, 1994; 1998; Chen & Liu, 2018), i.e., learning from an infinite stream of data, continuously integrates the newly learned knowledge without forgetting the old knowledge, which is also called lifelong learning (Silver & Mercer, 2002; Rannen et al., 2017; Aljundi et al., 2017; Chen & Liu, 2018; Chaudhry et al., 2019; Parisi et al., 2019), or incremental learning (Gepperth & Karaoguz, 2016; Shmelkov et al., 2017; Rebuffi et al., 2017; Aljundi et al., 2018; Chaudhry et al., 2018; Rosenfeld & Tsotsos, 2018; Zhang & Yang, 2022). According to whether the training data is labeled or not, continual learning can be divided into two categories: supervised continual learning (SCL), and unsupervised continual learning (UCL). SCL has been studied extensively in the past few years (Ratcliff, 1990; Li & Hoiem, 2017; Zenke et al., 2017; Buzzega et al., 2020; Lin et al., 2021; Arani et al., 2022). However, motivated by the practical need in real-world application scenarios, researchers have started to turn their attention to the unsupervised field: representation learning with unlabeled data on sequential tasks (i.e., UCL). UCL (Achille et al., 2018; Smith et al., 2019; Rao et al., 2019) aims to mitigate the catastrophic forgetting problem (McCloskey & Cohen, 1989) on a new task in an unsupervised way, which has a wide use in realistic application scenarios where unlabeled data is being produced over time.

Recent UCL methods (Lin et al., 2021; Madaan et al., 2021; Fini et al., 2022) have achieved promising performance by exploring various unsupervised strategies. However, most of them focus on utilizing a replay buffer to store previous data (i.e., rehearsal-based strategy), which needs much extra storage and thus limits their practical applications. For example, Lin et al. (2021) proposes a rehearsal method to solve the catastrophic forgetting problem in continual contrastive self-supervised learning. However, this method has to store a constant number of most consistent images per class to compute the intra-contrast (on old data), and it also requires an extra sample queue to store negative samples of old data to compute the inter-contrast (across old and new data), which is extremely cumbersome and storage-wasting. Madaan et al. (2021) introduces multiple techniques to address the catastrophic forgetting problem in UCL, but the two better techniques DER (Buzzega et al., 2020) and LUMP still depend on a replay buffer to mitigate forgetting by storing old data.

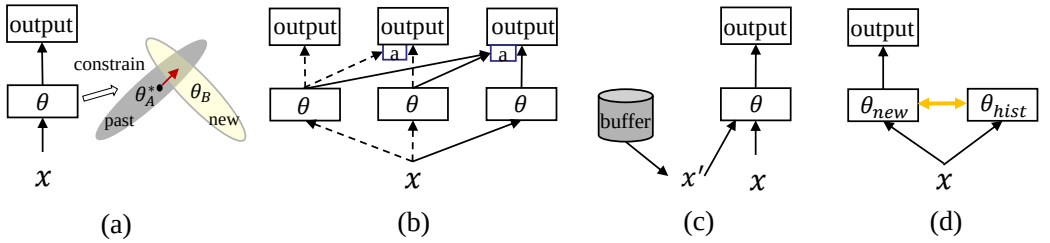

Figure 1: Schematic illustration of the three classic strategies and our proposed rememory-based strategy for continual learning. (a) **Regularization-based strategy** mitigates forgetting by adding a regularization term to the loss function to constrain the changes of model parameters. (b) **Architecture-based strategy** mitigates forgetting by expanding the model to provide a different set of parameters for each task (i.e., task-specific parameters). (c) **Rehearsal-based strategy** mitigates forgetting by storing and replaying selected samples of previous tasks with a memory buffer. (d) **Rememory-based strategy** proposed in this paper mitigates forgetting by the memory & remembering process between new-module (new) and hist-module (hist) instead of memory buffer.

To overcome the drawback of most recent UCL methods, based on unsupervised contrastive learning via SimSiam (Chen & He, 2021), we propose a novel rememory-based SimSiam (RM-SimSiam) method to reduce the dependency on replay buffer under the UCL setting. Analogous to the memory mechanism of the human brain (Poldrack et al., 2001; Shohamy & Wagner, 2008), the core idea of our RM-SimSiam is to store and remember the old knowledge with a data-free historical module (instead of replay buffer that stores old data directly). Specifically, our RM-SimSiam model mainly consists of two modules: hist-module (i.e., historical module) and new-module (see Figure 2). The hist-module is designed to store the historical average model of all previous models (i.e., the memory process) and then transfer the knowledge of the historical average model to the new-module (i.e., the rememory process) for retaining the previously learned knowledge. By such memory & rememory process, the old knowledge can be effectively consolidated (memorized) and remembered (rememorized) throughout the optimization trajectory, thus ensuring that RM-SimSiam can mitigate the catastrophic forgetting of the old knowledge when learning a new task. Note that the rememory-based strategy proposed in this work is a grand new and important strategy for continual learning, in addition to the other three classic strategies including regularization-based strategy, architecture-based strategy, and rehearsal-based strategy (see Figure 1).

Furthermore, to improve the rememory ability of RM-SimSiam, we devise an enhanced SimSiam-based contrastive loss by aligning the feature representations outputted by the historical and new models. Such alignment mechanism is different from that in the latest work (Fini et al., 2022) on UCL. Fini et al. (2022) exploits the distillation mechanism to align the representations of the current and past states by saving the model checkpoint of the past state. In contrast, we align the representations of the historical average model (of all previous models) and new model in each iteration process. Note that the largest difference between Fini et al. (2022) and our RM-SimSiam still lies in that the novel rememory process is included in our RM-SimSiam to learn a new task well while mitigating forgetting, but such rememory process is ignored in Fini et al. (2022).

We conduct extensive experiments on three benchmark datasets including S-CIFAR-10, S-CIFAR-100, and S-TINY-IMAGENET, following Madaan et al. (2021). The obtained results demonstrate the effectiveness of our RM-SimSiam under the UCL setting. Moreover, our RM-SimSiam can achieve further improvements when a replay buffer is used, showing that it is complementary to rehearsal-based methods. Finally, the experimental results on the out-of-distribution datasets demonstrate the superior generalization ability of our RM-SimSiam.

Our main contributions are three-fold: **(1)** We propose a novel rememory-based method termed RM-SimSiam for unsupervised continual learning by storing and remembering the old knowledge with a data-free historical module instead of replay buffer. **(2)** To effectively rememory the knowledge of previous tasks, we design a hist-module by storing the knowledge of previous models and transferring the knowledge of previous models to the new model. To further improve the rememory ability of our RM-SimSiam, we devise an enhanced SimSiam-based contrastive loss by aligning the representations outputted by the historical and new models. **(3)** Extensive experiments on three benchmarks show that our RM-SimSiam achieves new state-of-the-art under the UCL setting.

2 Related Work

Continual Learning. Different from traditional learning from data, continual learning mainly aims to continuously consolidate the learned knowledge from a stream of data. Below we introduce three classic strategies for continual learning. **(1) Regularization-Based Strategy (Figure 1(a)):** Earlier work LwF (Li & Hoiem, 2017) updates the model parameters by forcing the model prediction on the new task to approximate that on the old task (i.e., knowledge distillation (Hinton et al., 2015; Sarfraz et al., 2021)) only with the new data. EWC (Kirkpatrick et al., 2017) constrains the changes of model parameters that are important to the previous tasks by using a quadratic penalty to ensure that they do not deviate too far from the initial values. Similar to EWC, SI (Zenke et al., 2017) also evaluates the importance of model parameters and minimizes the changes of important parameters, but it differs from the EWC paradigm in that the parameters importance of old tasks and the new task is evaluated in a separate stage after training, which can ensure an online estimate of parameters importance. **(2) Rehearsal-Based Strategy (Figure 1(c)):** The idea of iCaRL (Rebuffi et al., 2017) is closer to LwF, the model parameters are updated by minimizing the distillation loss, but it retains a part of the representative old data for training, i.e., it is a combination of replay and distillation. GEM (Lopez-Paz & Ranzato, 2017) proposes a gradient fragment memory algorithm, which only updates the parameters of the new task and corrects the gradient update direction of the new task in an inequality-constrained manner. The recent method DER (Buzzega et al., 2020) mitigates forgetting by aligning the network output logits of the new model and that of the old model for the the old data throughout the optimization trajectory. Arani et al. (2022) achieves good performance by maintaining a large memory buffer to implement an interaction of dual semantic memories and episodic memory. **(3) Architecture-Based Strategy (Figure 1(b)):** There is a less commonly-used architecture-based strategy due to its very large storage requirement, such as PNN (Rusu et al., 2016), PackNet (Mallya & Lazebnik, 2018), HAT (Serra et al., 2018). Note that the research on these three classic strategies for continual learning is limited to the context of supervised learning, and we can easily extend them to the context of unsupervised learning like Madaan et al. (2021).

Unsupervised Representation Learning. Since data labeling is very expensive for supervised learning, researchers have focused on unsupervised learning with grand success in recent years. Most of unsupervised learning methods adopt contrastive learning, which minimizes the distance between anchor and positive samples, and maximizes the distance between anchor and negative samples. The most representative method Moco (He et al., 2020) takes two randomly-augmented views of one image as positive samples and that of the other images as negative samples, and maintains a continuously-updated negative sample queue to improve the consistency through a momentum encoder. Instead of using a queue to store negative samples, SimCLR (Chen et al., 2020) directly regards the other images in a batch as negative samples. Obviously, SimCLR needs a large batch size to learn well in the training process. Faced with these limitations, some negative-free methods are proposed. For example, BarlowTwins (Zbontar et al., 2021) learns image invariant features by forcing the cross-correlation matrix between different augmentations of an image to be close to the identity matrix. SimSiam (Chen & He, 2021) learns feature representations by minimizing the cosine-distance between different augmentations of an image, which is simple and effective.

Unsupervised Continual Learning. Recently, unsupervised continual learning (UCL) has started to receive much attention from the continual learning community. UCL aims to study the problem of learning unsupervised representations on sequential tasks. For example, Rao et al. (2019) proposes a Continual Unsupervised Representation Learning (CURL) method to learn unsupervised representation, combined with rehearsal-based methods to alleviate catastrophic forgetting. However, these methods are only suitable for some smaller digit-based datasets such as MNIST (LeCun et al., 2010) and Omniglot (Lake et al., 2011). Recently, Lin et al. (2021) focuses on contrastive learning, and solves its catastrophic forgetting problem by combining data rehearsal and knowledge distillation. However, this method has certain limitations and lacks the generalization to general unsupervised learning. Fini et al. (2022) proposes the Cassle strategy to mitigate forgetting through a designed prediction head, but the function of prediction head is different from that in our basic framework (SimSiam (Chen & He, 2021)), and the SimSiam loss is unsuitable for the Cassle strategy. LUMP (Madaan et al., 2021) extends multiple strategies from SCL to UCL, and moves towards general continual learning, which greatly promotes the development of UCL. Following LUMP, we introduce a novel rememory-based method for UCL, and the reported results in our main experiments (see Sec. 4.2) demonstrate that our proposed method significantly outperforms LUMP and even its performance is improved while greatly enhancing the model's generalization ability.

3 METHODOLOGY

3.1 PROBLEM DEFINITION

In this paper, we focus on the task-incremental learning setting for continual learning. Our main goal is to explore the ability of neural network to mimic the human brain memory mechanism, i.e., learning the knowledge of new tasks without forgetting the learned knowledge of previous tasks, which is a fundamental problem in continual learning. Under the task-incremental learning setting, we are given a sequence of tasks $T = \{T_1, T_2, ..., T_n\}$, where n denotes the number of tasks. Each task T_t ($1 \leq t \leq n$) from T has a task-specific training set $D_t = \{x_i, y_i\}_{i=1}^{N_t}$, where x_i denotes an image, y_i denotes the ground-truth class label of x_i, and N_t denotes the number of training samples. Given that D_t is drawn from the i.i.d. distribution $P_t(x, y)$, we assume that any pair of tasks T_t and T_{t+j} ($1 \leq j \leq n - t$) have different distributions: $P_t(x, y) \neq P_{t+j}(x, y)$. In addition, for each T_t, its validation and test sets can be defined similarly.

Since UCL is considered (but not SCL) in this paper, there is no labeled samples during training. That is, for each task T_t, it has an unlabeled training set $U_t = \{x_i\}_{i=1}^{N_t}$ with N_t training samples (but its validation and test sets have labeled samples). The learning process for UCL is thus given as follows: (1) The feature representations of the training samples are learned on the set of sequential tasks; (2) K-nearest neighbor (KNN) classifier (Wu et al., 2018) is performed on the validation set to obtain the classification accuracy for hyperparameter tuning; (3) The performance on the test set is evaluated based on KNN classifier, following the setup in (Chen et al., 2020; Zbontar et al., 2021; Chen & He, 2021).

3.2 SIMSIAM

SimSiam (Chen & He, 2021) is a simple yet effective method for unsupervised representation learning, which aims to learn the feature representations by minimizing the cosine-distance between two randomly-augmented views of an input image. Concretely, SimSiam mainly includes an encoder f and a predictor head h, just like the new-module in Figure 2. The encoder f consists of the backbone ResNet18 (He et al., 2016) (without pretraining), and the predictor head h consists of multilayer perceptron (MLP) layers. Given an input image x, the output of the encoder f is $z \triangleq f(x)$, and the output of the predictor h is $p \triangleq h(z) \triangleq h(f(x))$. For the two augmented views x_1 and x_2 of the input image x, SimSiam chooses to learn the feature representations by minimizing the cosine-distance between the output of one view's predictor (e.g., $x_1 \longrightarrow p_1 \triangleq h(f(x_1))$) and the output of the other view's encoder (e.g., $x_2 \longrightarrow z_2 \triangleq f(x_2)$) and vice versa.

According to Chen & He (2021), a symmetric contrastive loss L_{sim} is employed to learn more accurate representations (similar to dense sampling). The loss L_{sim} is defined as:

$$L_{sim} = \frac{1}{2}D(p_1, z_2) + \frac{1}{2}D(p_2, z_1), \tag{1}$$

$$D(p_1, z_2) = -\frac{p_1}{\|p_1\|_2} \cdot \frac{z_2}{\|z_2\|_2}, \tag{2}$$

where D is a cosine-distance function, and $\|\cdot\|_2$ is l_2-norm. Since a stop-gradient operation $\mathrm{sg}(\cdot)$ is imposed on z to prevent model collapse, L_{sim} is reformulated as:

$$L_{sim} = \frac{1}{2}D(p_1, \mathrm{sg}(z_2)) + \frac{1}{2}D(p_2, \mathrm{sg}(z_1)). \tag{3}$$

When SimSiam is applied to continual learning, given an input image $x_{i,t}$ from the task T_t, the symmetric contrastive loss L_{sim} is defined as:

$$L_{sim} = \frac{1}{2}D(p_{i,t}^1, \mathrm{sg}(z_{i,t}^2)) + \frac{1}{2}D(p_{i,t}^2, \mathrm{sg}(z_{i,t}^1)), \tag{4}$$

where the two augmented views of $x_{i,t}$ are $x_{i,t}^1$ and $x_{i,t}^2$, the encoder output $z_{i,t}^j \triangleq f(x_{i,t}^j)$ ($j = 1, 2$), and the predictor output $p_{i,t}^j \triangleq h(f(x_{i,t}^j))$ ($j = 1, 2$).

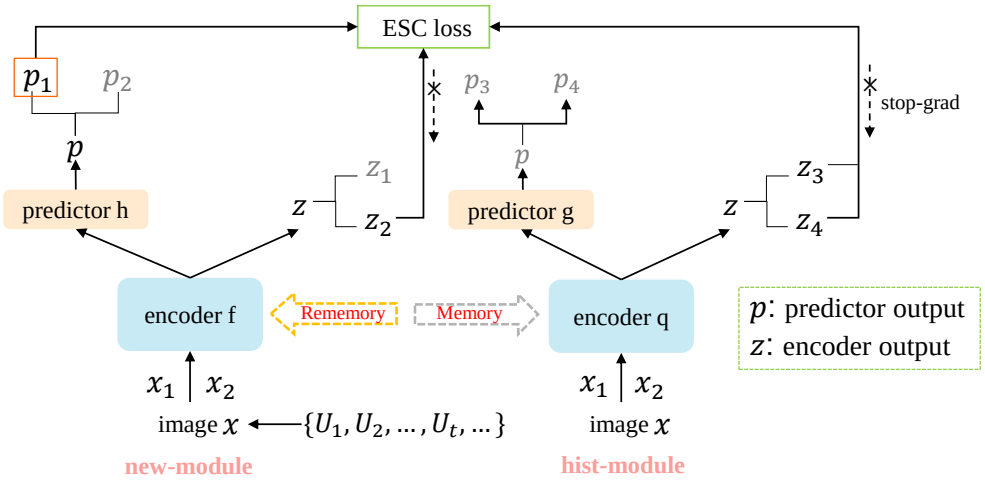

Figure 2: Overview of our RM-SimSiam. RM-SimSiam mainly consists of the new-module and hist-module. The rememory mechanism is applied between them to learn the new knowledge well while retaining the old knowledge. The enhanced SimSiam-based contrastive loss (ESC loss) for model optimization is defined by taking both the historical and new models into consideration.

3.3 RM-SimSiam

Inspired by the memory mechanism of human brain and based on unsupervised contrastive learning via SimSiam model, we propose the novel rememory-based SimSiam (RM-SimSiam) under the UCL setting. As illustrated in Figure 2, RM-SimSiam mainly has two modules: new-module and hist-module (historical module). Among them, the new-module is mainly used to learn the knowledge of the current new task (e.g., T_t), the hist-module is mainly used to retain the learned knowledge in the previous tasks (e.g., $T_1, T_2, ..., T_{t-1}$). With the proposed rememory mechanism and the enhanced SimSiam-based contrastive loss, our RM-SimSiam can learn new knowledge well while mitigating catastrophic forgetting. Below we describe these two key components in detail.

Rememory Mechanism for UCL. To mitigate the catastrophic forgetting problem under the UCL setting, we propose the rememory mechanism to consolidate (memory) and remember (rememory) the previously learned knowledge. Specifically, we design the hist-module to retain the old knowledge by storing the historical average model of all previous models (i.e., the memory process) and then transferring the knowledge of the historical average model to the new-module (i.e., the rememory process). As shown in Figure 2, in the new-module and hist-module, the encoders are respectively denoted as f (with parameters θ_f^e) and q (with parameters θ_q^e), and the predictor heads respectively as h (with parameters θ_h^p) and g (with parameters θ_g^p). To consolidate the learned knowledge of previous tasks, we update the parameters θ_q^e, θ_g^p of the hist-module by transferring the parameters θ_f^e, θ_h^p of the new-module, which is called the memory process. In turn, to remember the previously learned knowledge, we transfer the parameters of the hist-module to the new-module, which is called the rememory process. These two transfer processes constitute our rememory mechanism. Given the transfer coefficient m, the two transfer processes are uniformly defined as:

$$\theta_i^e = m \cdot \theta_i^e + (1-m) \cdot \theta_j^e, \;\; i,j \in \{f,q\}, \; i \neq j, \tag{5}$$

$$\theta_i^p = m \cdot \theta_i^p + (1-m) \cdot \theta_j^p, \;\; i,j \in \{h,g\}, \; i \neq j, \tag{6}$$

where the parameters θ_q^e, θ_g^p of the hist-module have no gradient back-propagation.

Enhanced SimSiam-based Contrastive Loss. Further, to improve the rememory ability of RM-SimSiam, we propose an enhanced SimSiam-based contrastive (ESC) loss by aligning the feature representations outputted by the historical and new models. Concretely, given an input image $x_{i,t}$, the new-module and hist-module take two randomly-augmented views $x_{i,t}^1, x_{i,t}^2$ of $x_{i,t}$ as inputs, and produce the corresponding encoder outputs $\{z_{i,t}^j\}$ and predictor outputs $\{p_{i,t}^j\}$ ($j = 1, 2$ for the new-module and $j = 3, 4$ for the hist-module), as shown in Figure 2. To better retain the previously learned knowledge, we add a new SimSiam-style contrastive loss L_{hist} on top of the original SimSiam loss L_{sim} given by Eq. (4). Formally, by taking the outputs of the two views in

the hist-module as guidance, we can define L_{hist} (with a similar form to L_{sim}) as follows:

$$L_{hist} = \frac{1}{2}D(p^1_{i,t}, z^3_{i,t}) + \frac{1}{2}D(p^3_{i,t}, z^1_{i,t}) + \frac{1}{2}D(p^1_{i,t}, z^4_{i,t}) + \frac{1}{2}D(p^4_{i,t}, z^1_{i,t}) + \frac{1}{2}D(p^2_{i,t}, z^3_{i,t}) + \frac{1}{2}D(p^3_{i,t}, z^2_{i,t}) + \frac{1}{2}D(p^2_{i,t}, z^4_{i,t}) + \frac{1}{2}D(p^4_{i,t}, z^2_{i,t}) + \frac{1}{2}D(p^3_{i,t}, z^4_{i,t}) + \frac{1}{2}D(p^4_{i,t}, z^3_{i,t}). \quad (7)$$

Noticing the non-gradient property of the hist-module, we further impose the stop-gradient operation $\mathrm{sg}(\cdot)$ on z. In this way, we can simplify the above contrastive loss L_{hist} as:

$$L_{hist} \triangleq \frac{1}{2}D(p^1_{i,t}, \mathrm{sg}(z^3_{i,t})) + \frac{1}{2}D(p^2_{i,t}, \mathrm{sg}(z^3_{i,t})) + \frac{1}{2}D(p^1_{i,t}, \mathrm{sg}(z^4_{i,t})) + \frac{1}{2}D(p^2_{i,t}, \mathrm{sg}(z^4_{i,t})). \quad (8)$$

By combining L_{sim} and L_{hist}, our enhanced SimSiam-based contrastive (ESC) loss is defined as:

$$L_{esc} = L_{sim} + \gamma L_{hist}, \quad (9)$$

where γ is the weight hyperparameter, and L_{sim} is the original SimSiam loss. The pseudocode of our full algorithm is given in Appendix A. Note that our proposed method can be combined with other contrastive learning methods like BarlowTwins (Zbontar et al., 2021) for UCL (see Appendix D).

4 Experiments

4.1 Experimental Setup

Datasets. Three classical datasets are selected for performance evaluation: **(1) SPLIT CIFAR-10** (S-CIFAR-10) (Krizhevsky et al., 2009) has a total of 10 classes with 60,000 images. Each class has 6,000 color images of $32 * 32$, of which 5,000 are used for training and 1,000 for testing. We split this dataset into 5 tasks, each of which contains 2 classes. **(2) SPLIT CIFAR-100** (S-CIFAR-100) (Krizhevsky et al., 2009) is composed of 100 classes. Each class has 600 color images of $32 * 32$, of which 500 are used for training and 100 for testing. We split this dataset into 20 tasks, each of which contains 5 classes. **(3) SPLIT TINY-IMAGENET** (S-TINY-IMAGENET) (Banerjee & Iyer, 2015) is a subset of ImageNet (Deng et al., 2009) (1,000 classes). Following Zenke et al. (2017); De Lange et al. (2021), we only use the first 100 classes for continual learning, each of which has 500 color images for training and 50 images for testing. The task split is the same as that of S-CIFAR-100. The image size is $64 * 64$ in this dataset. Overall, all classes of each dataset are kept in fixed order for sequential training across three independent runs.

Evaluation Metrics. To evaluate the model performance under the UCL setting, the two metrics average accuracy and average forgetting are reported, following De Lange et al. (2021); Chen & He (2021). **(1) Average accuracy.** Let $a_{t,i}$ denote the test accuracy on task T_i ($1 \leq i \leq t$) after learning the current task T_t. The average accuracy is defined as: $A_t = \frac{1}{t}\sum_{i=1}^{t} a_{t,i}$, which refers to the test average accuracy on all learned tasks after learning the current task T_t. **(2) Average forgetting.** Forgetting defines the difference between the maximum accuracy obtained by the learned tasks (e.g., $t' = 1, 2, 3$) on previous task i ($1 \leq i \leq t-1$) when learning the current task T_t (e.g., $t = 3$) and the accuracy obtained by the current task T_t on task i. Thus, the average forgetting is formulated as: $F_t = \frac{1}{t-1}\sum_{i=1}^{t-1} \max_{t' \in \{1,\ldots,t\}}(a_{t',i} - a_{t,i})$, which refers to the average of forgetting on the previous tasks after learning the current task. In the following experiments, the average accuracy (acc) and average forgetting (fg) of the final model are reported when learning all tasks (i.e., $t = n$).

Implementation Details. Our RM-SimSiam adopts the Stochastic Gradient Descent (SGD) optimizer, with the learning rate $\eta = 0.03$ for S-CIFAR-10/S-CIFAR-100 and $\eta = 0.035$ for S-TINY-IMAGENET. We set the batch size to 128. Following Madaan et al. (2021), we average the results (acc and fg) over three independent runs as the final results. During the training phase, we perform the common augmentation operations (e.g., random crops, horizontal flips, and color jittering) on the training set. We split the training set into two parts with the ratio $9:1$, i.e., 90% of the whole training set for training and the rest 10% for validation. During the test phase, the raw input images are directly used for evaluation. We set the transfer coefficient between two modules for our RM-SimSiam to $m = 0.99$ in all the experiments. The weight hyperparameter in the loss function of our RM-SimSiam is set to $\gamma = 1$. To explore the complementarity between the rehearsal-based method and our RM-SimSiam, we combine our RM-SimSiam with the Mixup strategy (Zhang et al., 2017). The old data is stored and replayed in a memory buffer (buffer size 256) by adopting the reservoir sampling to guarantee the same probability for each sample following Buzzega et al. (2020). The interpolation hyperparameter α for Mixup is set to $0.3, 0.26, 0.42$ for S-CIFAR-10, S-CIFAR-100 and S-TINY-IMAGENET, respectively. The source code will be released soon.

Table 1: Comparison to the state-of-the-arts under the UCL setting in terms of average accuracy and average forgetting over three independent runs. 'acc' refers to average accuracy, and 'fg' refers to average forgetting. The standard deviation is given in brackets. All UCL methods (with the same backbone ResNet18) are trained from scratch. * denotes our RM-SimSiam without buffer.

Method	S-CIFAR-10		S-CIFAR-100		S-TINY-IMAGENET	
	acc (↑)	fg (↓)	acc (↑)	fg (↓)	acc (↑)	fg (↓)
FINETUNE	90.11 (±0.12)	5.42 (±0.08)	75.42 (±0.78)	10.19 (±0.37)	71.07 (±0.20)	9.48 (±0.56)
PNN (Rusu et al., 2016)	90.93 (±0.22)	–	66.58 (±1.00)	–	62.15 (±1.35)	–
SI (Zenke et al., 2017)	92.75 (±0.06)	1.81 (±0.21)	80.08 (±1.30)	5.54 (±1.30)	72.34 (±0.42)	8.26 (±0.64)
DER (Buzzega et al., 2020)	91.22 (±0.30)	4.63 (±0.26)	77.27 (±0.30)	9.31 (±0.09)	71.90 (±1.44)	8.36 (±2.06)
LUMP (Madaan et al., 2021)	91.00 (±0.40)	2.92 (±0.53)	82.30 (±1.35)	4.71 (±1.52)	76.66 (±2.39)	3.54 (±1.04)
Cassle (Fini et al., 2022)	90.84 (±0.13)	2.29 (±0.23)	76.46 (±1.02)	3.05 (±0.87)	71.99 (±0.46)	3.34 (±0.52)
RM-SimSiam* (ours)	91.22 (±0.12)	4.15 (±0.18)	78.48 (±0.31)	4.09 (±0.99)	72.25 (±0.06)	4.51 (±0.04)
RM-SimSiam (ours)	**93.07 (±0.13)**	**1.36 (±0.10)**	**83.26 (±0.30)**	**2.73 (±0.42)**	**77.10 (±0.16)**	**2.67 (±0.01)**
MULTITASK	95.76 (±0.08)	–	86.31 (±0.38)	–	82.89 (±0.49)	–

Table 2: Comparison to the state-of-the-arts on the out-of-distribution (OOD) datasets. All UCL methods are trained on S-CIFAR-10 or S-CIFAR-100, and then directly tested on the OOD datasets.

IN-CLASS	S-CIFAR-10				S-CIFAR-100			
OUT-OF-CLASS	MNIST	FMNIST	SVHN	CIFAR-100	MNIST	FMNIST	SVHN	CIFAR-10
FINETUNE	89.23 (±0.99)	80.05 (±0.34)	49.66 (±0.81)	34.52 (±0.12)	85.99 (±0.86)	76.90 (±0.11)	50.09 (±1.41)	57.15 (± 0.96)
SI (Zenke et al., 2017)	93.72 (±0.58)	82.50 (±0.51)	**57.88 (±0.16)**	36.21 (±0.69)	91.50 (±1.26)	80.57 (±0.93)	54.07 (±2.73)	60.55 (±2.54)
DER (Buzzega et al., 2020)	88.35 (±0.82)	79.33 (±0.62)	48.83 (±0.55)	30.68 (±0.36)	87.96 (±2.04)	76.21 (±0.63)	47.70 (±0.94)	56.26 (±0.16)
LUMP (Madaan et al., 2021)	91.03 (±0.22)	80.78 (±0.88)	45.18 (±1.57)	31.17 (±1.83)	91.76 (±1.17)	81.61 (±0.45)	50.13 (±0.71)	63.00 (±0.53)
Cassle (Fini et al., 2022)	89.81 (±0.32)	80.98 (±0.03)	50.64 (±0.56)	34.25 (±1.13)	88.87 (±0.45)	81.30 (±0.45)	51.04 (±0.01)	59.46 (±1.62)
RM-SimSiam (ours)	**94.32 (±0.26)**	**83.33 (±0.21)**	53.35 (±2.69)	**42.02 (±0.37)**	**94.96 (±0.21)**	**83.29 (±0.19)**	**60.37 (±1.72)**	**69.16 (±0.17)**
MULTITASK	90.69 (±0.13)	80.65 (±0.42)	47.67 (±0.45)	39.55 (±0.18)	90.35 (±0.24)	81.11 (±1.86)	52.20 (±0.61)	70.19 (±0.15)

4.2 Main Results

We compare our proposed RM-SimSiam against other state-of-the-art methods under the UCL setting on the three benchmark datasets. These state-of-the-art methods for UCL are composed of: **(1)** the unsupervised representation learning (URL) approach SimSiam (Chen & He, 2021) with various anti-forgetting strategies including one architecture-based strategy (PNN (Rusu et al., 2016)), one regularization-based strategy (SI (Zenke et al., 2017)), and two rehearsal-based strategies (DER (Buzzega et al., 2020), LUMP (Madaan et al., 2021)); **(2)** another URL approach BarlowTwins (Zbontar et al., 2021) with the Cassle strategy (Fini et al., 2022). For fair comparison, both RM-SimSiam with memory buffer (denoted as RM-SimSiam) and RM-SimSiam without memory buffer (denoted as RM-SimSiam*) are considered. Note that FINETUNE is the lower bound which denotes training sequentially on all tasks without any anti-forgetting strategies, while MULTITASK is the upper bound which denotes training on all tasks as a whole (but not sequentially).

The comparative results in terms of average accuracy and average forgetting (over three independent runs) are shown in Table 1. It can be observed that: **(1)** Our RM-SimSiam without memory buffer (i.e., RM-SimSiam*) leads to better results than most of the other UCL methods on all the three benchmark datasets, demonstrating the effectiveness of our RM-SimSiam under the UCL setting. **(2)** When the memory buffer is used exactly the same as DER and LUMP, our RM-SimSiam beats all the other UCL methods and achieves new state-of-the-art results on all the three benchmark datasets for UCL. This indicates that our proposed RM-SimSiam is indeed complementary to the rehearsal-based strategy and provides a new perspective to mitigate forgetting in UCL. **(3)** Our RM-SimSiam outperforms the latest rehearsal-based method LUMP (Madaan et al., 2021) by 0.44% – 2.07% on accuracy and by 0.87% – 1.98% on forgetting, which provides direct evidence that our proposed rememory mechanism is crucial for learning the new task well while mitigating the forgetting problem under the UCL setting. **(4)** The accuracy margins between MULTITASK and our RM-SimSiam range from 2.69% to 5.79%. This suggests that there is still room for improvement in the research on UCL and other more advancing methods need to be further explored.

Table 2 shows the comparative results on the out-of-distribution (OOD) datasets. All UCL methods (with the same backbone ResNet18 (He et al., 2016)) are first trained on S-CIFAR-10 or S-CIFAR-100, and then directly tested on the OOD datasets. Following Madaan et al. (2021), the OOD evaluation is performed on MNIST (LeCun et al., 2010), Fashion-MNIST (FMNIST) (Xiao et al., 2017), SVHN (Netzer et al., 2011), CIFAR-100 (or CIFAR-10) (Krizhevsky et al., 2009), respec-

Table 3: Ablation study results for our full RM-SimSiam on S-CIFAR-10 and S-CIFAR-100 under the UCL setting. Notations: RM – the rememory mechanism; Hist – the extra contrastive loss L_{hist} defined based on the historical module (hist-module).

Method	S-CIFAR-10		S-CIFAR-100	
	acc (↑)	fg (↓)	acc (↑)	fg (↓)
Base (SimSiam)	90.16 (±0.24)	5.85 (±0.32)	75.51 (±0.70)	10.70 (±0.83)
Base+Mixup	90.40 (±0.18)	2.47 (±0.08)	77.89 (±0.77)	6.97 (±0.67)
Base+Mixup+RM	91.10 (±0.21)	1.67 (±0.41)	80.29 (±0.19)	4.24 (±0.45)
Base+Mixup+Hist	92.49 (±0.19)	1.96 (±0.26)	82.26 (±0.22)	3.91 (±0.26)
Base+Mixup+RM+Hist (full)	**93.07 (±0.13)**	**1.36 (±0.10)**	**83.26 (±0.30)**	**2.73 (±0.42)**

Table 4: Effect of m on our RM-SimSiam.

m	S-CIFAR-10		S-CIFAR-100	
	acc (↑)	fg (↓)	acc (↑)	fg (↓)
0.9	92.48 (±0.20)	2.25 (±0.37)	82.27 (±0.10)	3.25 (±0.12)
0.99	**93.07 (±0.13)**	1.36 (±0.10)	**83.26 (±0.30)**	2.73 (±0.42)
0.999	92.46 (±0.18)	1.31 (±0.22)	82.64 (±0.20)	**1.47 (±0.09)**
0.9999	91.23 (±0.11)	**0.45 (±0.14)**	80.67 (±0.32)	1.50 (±0.55)

Table 5: Effect of α on our RM-SimSiam.

α	S-CIFAR-10		S-CIFAR-100	
	acc (↑)	fg (↓)	acc (↑)	fg (↓)
0.22	92.60 (±0.13)	2.39 (±0.11)	82.87 (±0.33)	2.90 (±0.24)
0.26	92.72 (±0.02)	1.78 (±0.35)	**83.26 (±0.30)**	2.73 (±0.42)
0.30	**93.07 (±0.13)**	**1.36 (±0.10)**	82.62 (±0.09)	**2.56 (±0.46)**
0.34	92.73 (±0.14)	2.00 (±0.08)	82.58 (±0.16)	2.68 (±0.09)

tively. More details of the OOD experiments are provided in Appendix C. From Table 2, we have the following observations: **(1)** Our RM-SimSiam clearly outperforms the state-of-the-art methods (including SI and LUMP) according to the average performance over all tasks. Particularly, our RM-SimSiam beats the second best method SI (Zenke et al., 2017) on most tasks (except the task of training on S-CIFAR-10 followed by testing on SVHN). The obtained improvements on the OOD datasets show the superior generalization ability of our RM-SimSiam when unseen data distributions are encountered. **(2)** Our RM-SimSiam leads to remarkable improvements over MULTITASK on most tasks, and similar finding can also be obtained for SI. The improvements over MULTITASK indicate that the latest UCL methods tend to have better generalization ability than MULTITASK under the OOD setting (i.e., MULTITASK is not the upper bound for the OOD evaluation).

4.3 Further Evaluation

Ablation Study. To demonstrate the contribution of each key component (see Figure 2) of our full RM-SimSiam, we conduct ablation study on S-CIFAR-10 and S-CIFAR-100. We take SimSiam (Chen & He, 2021) as the first baseline (denoted as Base). On the basis of Base or SimSiam, we add the Mixup strategy to form the second baseline (denoted as Base+Mixup). Further, we add other key components including the rememory mechanism (RM) and the extra loss L_{hist} (Hist), which together make up our full RM-SimSiam. Therefore, four simplified versions of our full RM-SimSiam are included in the ablation study: (i) Base – SimSiam; (ii) Base+Mixup – SimSiam with the Mixup strategy; (iii) Base+Mixup+RM – SimSiam with the Mixup strategy and the rememory mechanism (RM); (iv) Base+Mixup+Hist – SimSiam with the Mixup strategy and the extra contrastive loss L_{hist}. Note that our full RM-SimSiam can be denoted as Base+Mixup+RM+Hist. The ablation study results in Table 3 demonstrate that: **(1)** The Mixup strategy leads to improvements over Base (SimSiam), due to the use of the old data from the memory buffer. **(2)** Our rememory mechanism brings further improvements on both accuracy and forgetting (see Base+Mixup+RM vs. Base+Mixup). This suggests that our rememory mechanism is complementary to the rehearsal-based method based on Mixup. **(3)** When the extra loss L_{hist} is added, we can see significant improvements over Base+Mixup, which indicates that L_{hist} has important effect on the model performance. **(4)** The combination of RM and L_{hist} yields further improvements, showing their complementarity under the UCL setting. **(5)** Our full RM-SimSiam achieves significant improvements over Base+Mixup, which means that we have made sufficient contributions by devising new rememory mechanism and enhanced SimSiam-based contrastive loss for UCL.

Effect of Hyperparameters. We conduct experiments on S-CIFAR-10 and S-CIFAR-100 to study the impact of two important hyperparameters m and α on the performance of our RM-SimSiam. Keep in mind that m is the transfer coefficient of the rememory mechanism, and α affects the distribution of interpolated coefficient λ of the Mixup strategy. Firstly, we explore the hyperparameter $m \in \{0.9, 0.99, 0.999, 0.9999\}$ while fixing the hyperparameter α (i.e., 0.30 and 0.26 respectively for S-CIFAR-10 and S-CIFAR-100). Table 4 shows the effect of m on the performance of our RM-SimSiam. We can see that: when m is set to 0.99, our method provides the highest accuracies on

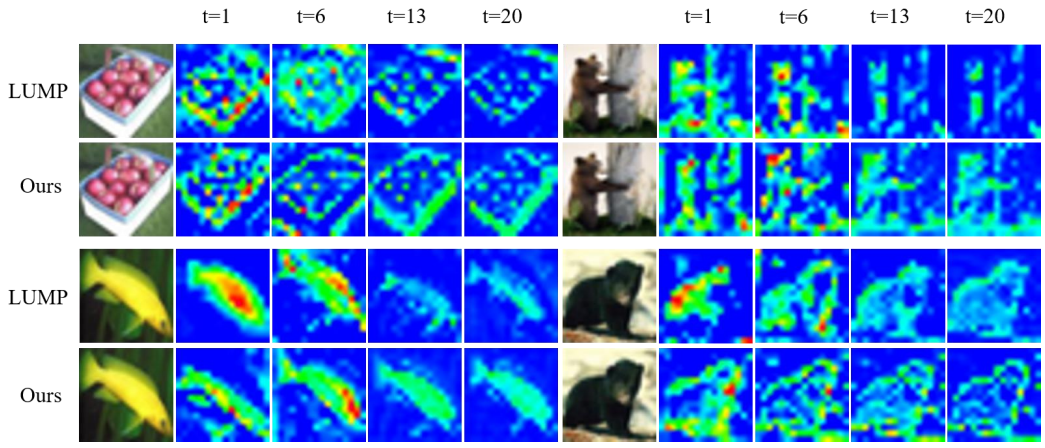

Figure 3: Visualization examples of feature maps from (the last layer of) the second block of the backbone ResNet18 when LUMP and our RM-SimSiam are being trained sequentially across all 20 tasks of S-CIFAR-100 (but only task 1, 6, 13 and 20 are shown for conciseness). The input images are randomly selected from the test set of task 1.

the two datasets; when m is larger (0.999 or 0.9999), the forgetting is lowered clearly but the accuracy is decreased. Considering the trade-off between accuracy and forgetting, we thus select m with the median value 0.99. Secondly, with $m = 0.99$ fixed, we further explore the hyperparameter $\alpha \in \{0.22, 0.26, 0.30, 0.34\}$. We do not try other values of α, since larger values tend to cause bad results on the two datasets. Table 5 shows the effect of α on the performance of our RM-SimSiam. We can see that: when α takes the values of 0.26 and 0.30, the model performance is relatively better. Particularly, on S-CIFAR-10, the model performance is obviously better when $\alpha = 0.30$, and on S-CIFAR-100, the model performance is slightly better when $\alpha = 0.26$ (with the highest accuracy). Therefore, we select $\alpha = 0.30$ on S-CIFAR-10 and $\alpha = 0.26$ on S-CIFAR-100 in this paper.

Visualization Results. To directly demonstrate the effectiveness of our RM-SimSiam under the UCL setting, we provide several visualization examples of feature maps from (the last layer of) the second block of the backbone ResNet18 in Figure 3, where the input images are randomly selected from the test set of task 1. The backbone ResNet18 is being trained sequentially across all the 20 tasks of S-CIFAR-100 (but only task 1, 6, 13 and 20 are shown for conciseness) by the state-of-the-art LUMP (Madaan et al., 2021) and our RM-SimSiam. We can clearly observe that our RM-SimSiam can better locate the important areas of the objects and represent the key visual features more stably across sequential tasks as compared with LUMP. For example, our RM-SimSiam can accurately identify the location of apples and the apple box (even the thickness of the box) in the top-left example, and pays more attention to the multiple objects, better capturing the visual features of an animal holding a stake in the top-right example. In the bottom-left example, our RM-SimSiam focuses on the shape, and can represent the visual features of fish tails and whiskers stably. In the bottom-right example, our RM-SimSiam can represent the outline features of objects more clearly and consistently. Overall, these visualization results show that the feature maps outputted by our RM-SimSiam have less degradation during sequential training, i.e., our RM-SimSiam forgets slower than LUMP. More visualization results are given in Appendix H.

5 Conclusion

In this paper, we propose a novel rememory-based method termed RM-SimSiam for unsupervised continual learning by storing and remembering the old knowledge with a data-free historical module instead of replay buffer. Specifically, to effectively rememory the knowledge of previous tasks, we design a hist-module by storing the knowledge of previous models and transferring the knowledge of previous models to the new model. Moreover, to further improve the rememory ability of our RM-SimSiam, we devise an enhanced SimSiam-based contrastive loss by aligning the representations outputted by the historical and new models. Extensive experiments on three benchmarks demonstrate the effectiveness of our RM-SimSiam in mitigating the catastrophic forgetting under the UCL setting. The experiments on the out-of-distribution datasets further demonstrate the superior generalization ability of our RM-SimSiam in continual learning.

REFERENCES

Alessandro Achille, Tom Eccles, Loic Matthey, Chris Burgess, Nicholas Watters, Alexander Lerchner, and Irina Higgins. Life-long disentangled representation learning with cross-domain latent homologies. *Advances in Neural Information Processing Systems (NeurIPS)*, 31:9895–9905, 2018.

Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3366–3375, 2017.

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

Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. Learning fast, learning slow: A general continual learning method based on complementary learning system. *arXiv preprint arXiv:2201.12604*, 2022.

Arijit Banerjee and Vignesh Iyer. Cs231n project report-tiny imagenet challenge, 2015.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:15920–15930, 2020.

Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *European Conference on Computer Vision (ECCV)*, pp. 532–547, 2018.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *International Conference on Learning Representations (ICLR)*, 2019.

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

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15750–15758, 2021.

Zhiyuan Chen and Bing Liu. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 12(3):1–207, 2018.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44(7):3366–3385, 2021.

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

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

Alexander Gepperth and Cem Karaoguz. A bio-inspired incremental learning architecture for applied perceptual problems. *Cognitive Computation*, 8(5):924–934, 2016.

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

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

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

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *National Academy of Sciences*, 114(13):3521–3526, 2017.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*, 1(4), 2009.

Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. One-shot learning of simple visual concepts. In *Annual Meeting of the Cognitive Science Society*, volume 33, pp. 2568–2573, 2011.

Yann LeCun, Corinna Cortes, and CJ Burges. MNIST handwritten digit database. AT&T labs [online]. *yann. lecun. com/exdb/mnist*, 2010.

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

Zhiwei Lin, Yongtao Wang, and Hongxiang Lin. Continual contrastive self-supervised learning for image classification. *arXiv preprint arXiv:2107.01776*, 2021.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 30:6470–6479, 2017.

Divyam Madaan, Jaehong Yoon, Yuanchun Li, Yunxin Liu, and Sung Ju Hwang. Representational continuity for unsupervised continual learning. In *International Conference on Learning Representations (ICLR)*, 2021.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7765–7773, 2018.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

Russell A Poldrack, Jill Clark, EJet al Paré-Blagoev, Daphna Shohamy, J Creso Moyano, Catherine Myers, and Mark A Gluck. Interactive memory systems in the human brain. *Nature*, 414(6863): 546–550, 2001.

Amal Rannen, Rahaf Aljundi, Matthew B Blaschko, and Tinne Tuytelaars. Encoder based lifelong learning. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 1320–1328, 2017.

Dushyant Rao, Francesco Visin, Andrei Rusu, Razvan Pascanu, Yee Whye Teh, and Raia Hadsell. Continual unsupervised representation learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 32:7647–7657, 2019.

Roger Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological Review*, 97(2):285, 1990.

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

Mark Bishop Ring. *Continual Learning in Reinforcement Environments*. PhD thesis, University of Texas at Austin, USA, 1994. UMI Order No. GAX95-06083.

Mark Bishop Ring. Child: A first step towards continual learning. In *Learning to Learn*, pp. 261–292. Springer, 1998.

Amir Rosenfeld and John K Tsotsos. Incremental learning through deep adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 42(3):651–663, 2018.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Fahad Sarfraz, Elahe Arani, and Bahram Zonooz. Knowledge distillation beyond model compression. In *International Conference on Pattern Recognition (ICPR)*, pp. 6136–6143, 2021.

Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning (ICML)*, pp. 4528–4537, 2018.

Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning (ICML)*, pp. 4548–4557, 2018.

Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. Incremental learning of object detectors without catastrophic forgetting. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 3400–3409, 2017.

Daphna Shohamy and Anthony D Wagner. Integrating memories in the human brain: hippocampal-midbrain encoding of overlapping events. *Neuron*, 60(2):378–389, 2008.

Daniel L Silver and Robert E Mercer. The task rehearsal method of life-long learning: Overcoming impoverished data. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pp. 90–101, 2002.

James Smith, Seth Baer, Zsolt Kira, and Constantine Dovrolis. Unsupervised continual learning and self-taught associative memory hierarchies. In *ICLR Workshop on Learning from Limited Labeled Data*, 2019.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3733–3742, 2018.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

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

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning (ICML)*, pp. 3987–3995, 2017.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609, 2022.

A FULL ALGORITHM FOR RM-SIMSIAM

To make the idea of our RM-SimSiam clearer, we provide the pseudocode of the full algorithm for RM-SimSiam in Alg. 1. Note that the Mixup strategy (see Appendix G) is applied in this algorithm. Specifically, Mixup denotes an interpolation between the instances of current task and the instances of previous tasks. That is, the instances of previous tasks are stored and replayed from the memory buffer M by the reservoir sampling algorithm (see Alg. 2), thus ensuring that the instances are taken with the same probability.

Algorithm 1 Unsupervised Continual Learning with RM-SimSiam

Input: the memory buffer M, the dataset U
the new-module with parameters θ_{new}
the hist-module with parameters θ_{hist}
hyperparameters α and m, the learning rate η
Output: the learned θ^*_{new}
$M \longleftarrow \{\}$
for x in U **do**
$\theta_{hist} \longleftarrow \theta_{new}$ ▷ Initialize the hist-module
$(x_1^m, x_2^m) \longleftarrow \text{sample}(M)$
$x_1, x_2 \longleftarrow \text{augment}(x)$
$\lambda \longleftarrow \text{numpy.random.beta}(\alpha, \alpha)$
$\hat{x}_1 \longleftarrow \lambda \cdot x_1 + (1-\lambda) \cdot x_1^m$
$\hat{x}_2 \longleftarrow \lambda \cdot x_2 + (1-\lambda) \cdot x_2^m$
$z_1, z_2 \longleftarrow f_\theta(\hat{x}_1), f_\theta(\hat{x}_2)$ ▷ Compute the outputs of the new-module
$p_1, p_2 \longleftarrow h_\theta(z_1), h_\theta(z_2)$
$z_3, z_4 \longleftarrow q_\theta(\hat{x}_1), q_\theta(\hat{x}_2)$ ▷ Compute the outputs of the hist-module
$p_3, p_4 \longleftarrow g_\theta(z_3), g_\theta(z_4)$
$\theta_{hist} \longleftarrow m \cdot \theta_{hist} + (1-m) \cdot \theta_{new}$ ▷ Update the hist-module
$\theta_{new} \longleftarrow m \cdot \theta_{new} + (1-m) \cdot \theta_{hist}$ ▷ Reverse update the new-module
$\theta_{new} = \theta_{new} - \eta \cdot \nabla_{\theta_{new}} L_{esc}$
$M \longleftarrow \text{reservoir}(x, \hat{x}_2)$
end for
return the found best θ^*_{new}

Algorithm 2 Reservoir Sampling Algorithm

Input: the memory buffer M
the number of seen examples N
a sample pair (x_1, x_2)
Output: the updated M
if $|M| > N$ **then**
$M[N] \leftarrow (x_1, x_2)$
else
$j = \text{randomInteger}(\min = 0, \max = N)$ ▷ Generate random integers
if $j < |M|$ **then**
$M[j] \leftarrow (x_1, x_2)$
end if
end if
return the updated M

B DISCUSSION ON THE DIFFERENCE FROM PROGRESS & COMPRESS

Although the rememory and memory phases in the proposed method are very similar to the progress and compress phases in Schwarz et al. (2018), the proposed method is quite different in that the parameters are directly transferred between the old and new models, while either predicted class probabilities (in case of supervised learning) or policies/values (in case of reinforcement learning) are aligned/transferred between the old and new tasks (or models). From this viewpoint,

Table 6: Comparison to the state-of-the-arts under the UCL setting. 'acc' refers to average accuracy, 'fg' refers to average forgetting. 'URL' refers to unsupervised representation learning. Different URL approaches (BarlowTwins and SimSiam) are deployed for UCL.

Method	URL	S-CIFAR-10		S-CIFAR-100		S-TINY-IMAGENET	
		acc (↑)	fg (↓)	acc (↑)	fg (↓)	acc (↑)	fg (↓)
FINETUNE	BarlowTwins	87.72 (±0.32)	4.08 (±0.56)	71.97 (±0.54)	9.45 (±1.01)	66.28 (±1.23)	8.89 (±0.66)
PNN (Rusu et al., 2016)	BarlowTwins	87.52 (±0.33)	–	57.93 (±2.98)	–	48.70 (±2.59)	–
SI (Zenke et al., 2017)	BarlowTwins	90.21 (±0.08)	2.03 (±0.22)	75.04 (±0.63)	7.43 (±0.67)	56.96 (±1.48)	17.04 (±0.89)
DER (Buzzega et al., 2020)	BarlowTwins	88.67 (±0.24)	2.41 (±0.26)	73.48 (±0.53)	7.98 (±0.29)	68.56 (±1.47)	7.87 (±0.44)
LUMP (Madaan et al., 2021)	BarlowTwins	90.31 (±0.30)	**1.13 (±0.18)**	80.24 (±1.04)	3.53 (±0.83)	72.17 (±0.89)	**2.43 (±1.00)**
Cassle (Fini et al., 2022)	BarlowTwins	90.84 (±0.13)	2.29 (±0.23)	76.46 (±1.02)	3.05 (±0.87)	71.99 (±0.46)	3.34 (±0.52)
Ours	BarlowTwins	**91.65 (±0.21)**	1.32 (±0.22)	**81.19 (±0.23)**	**1.85 (±0.33)**	**75.62 (±0.07)**	2.90 (±0.40)
Ours	SimSiam	93.07 (±0.13)	1.36 (±0.10)	83.26 (±0.30)	2.73 (±0.42)	77.10 (±0.16)	2.67 (±0.01)
MULTITASK	BarlowTwins	95.48 (±0.14)	–	87.16 (±0.52)	–	82.42 (±0.74)	–

Schwarz et al. (2018) still belongs to the traditional regularization-based methods, while the proposed method provides a (somewhat) new direction for continual learning. Note that further improvements achieved by our enhanced SimSiam-based contrast loss for knowledge aligning actually demonstrate the complementarity of rememory and regularization. In addition, the proposed method is devised for unsupervised continual learning, while Schwarz et al. (2018) is more suitable for supervised/reinforcement continual learning.

C More Details of Out-of-Distribution Experiments

We perform the evaluation on four out-of-distribution (OOD) datasets to show the generalization ability of our proposed RM-SimSiam. These OOD datasets are MNIST (LeCun et al., 2010), Fashion-MNIST (FMNIST) (Xiao et al., 2017), SVHN (Netzer et al., 2011), and CIFAR-10 (or CIFAR-100) (Krizhevsky et al., 2009). Concretely, (1) **MNIST** is a gray-scale digit-base dataset, which has a total of 10 classes with 70,000 images of the size $28 * 28$. Among them, 60,000 are used for training and 10,000 for testing. (2) **FMNIST** Similar to the MNIST dataset, FMNIST has the same number and size of gray-scale images, and the same training/test set split. Differently, FMNIST contains 10 categories of images including t-shirt, trouser, pullovers, skirts and sandals, etc. (3) **SVHN** (Street View House Number) is a digit-base dataset, where each image of the size $32 * 32 * 3$ contains a set of '0-9' Arabic numerals (10 categories as MNIST). There are 73,257 digits in the training set, 26,032 digits in the test set, and 531,131 additional digits. (4) **CIFAR-10/CIFAR-100** CIFAR-10 has a total of 10 classes with 60,000 images of the size $32 * 32 * 3$. 50,000 are used for training and 10,000 for testing. CIFAR-100 has a total 60,000 color images of the same size, with the same training/test split as the CIFAR-10 dataset, but have 100 categories. For fair comparison, all UCL methods are first trained on the S-CIFAR-10 (or S-CIFAR-100 dataset) across three independent runs (with different random initializations), and then directly evaluated on the test set of the four OOD datasets. The average accuracies over the three independent runs obtained by all UCL methods on each OOD dataset have been reported in Table 2 of the main paper.

D Alternative Contrastive Learning Approach

In the main paper, we employ the unsupervised representation learning (URL) approach SimSiam (Chen & He, 2021) to conduct extensive experiments and make comparisons with the-state-of-arts (see Table 1), demonstrating the effectiveness of our proposed rememory-based (RM) strategy. In the following, we provide another implementation that employs the URL approach BarlowTwins (Zbontar et al., 2021) to conduct experiments and make comparisons with the-state-of-arts (the setting of hyperparameters and batch size is the same as the experiments of RM-SimSiam), so that the effectiveness of our proposed RM strategy can be further validated. Here, the RM strategy applied to BarlowTwins with memory buffer (buffer size 256) is denoted as ours (with BarlowTwins). Similarly, the RM strategy applied to SimSiam with the same memory buffer size (i.e., RM-SimSiam) is denoted as ours (with SimSiam), which is still considered for extensive comparison. The definitions of FINETUNE and MULTITASK are the same as in the main paper.

The comparative results in terms of average accuracy and average forgetting (over three independent runs) are shown in Table 6. It can be observed that: (1) Our proposed RM strategy still obtains the best performance compared to other classic strategies on the basis of BarlowTwins. This further

Table 7: Comparison to the URL method BYOL (Grill et al., 2020) on S-CIFAR-10 and S-CIFAR-100 under the UCL setting. * denotes our RM-SimSiam without replay buffer.

Method	URL	S-CIFAR-10		S-CIFAR-100	
		acc (↑)	fg (↓)	acc (↑)	fg (↓)
BYOL+FINETUNE	BYOL	89.67 (±0.22)	5.11 (±0.34)	74.84 (±0.78)	5.92 (±0.61)
BYOL+Mixup	BYOL	92.26 (±0.42)	3.37 (±0.44)	81.30 (±0.50)	4.35 (±0.62)
RM-SimSiam* (ours)	SimSiam	91.22 (±0.12)	4.15 (±0.18)	78.48 (±0.31)	4.09 (±0.99)
RM-SimSiam (ours)	SimSiam	**93.07 (±0.13)**	**1.36 (±0.10)**	**83.26 (±0.30)**	**2.73 (±0.42)**

Table 8: Fair memory comparison on S-CIFAR-10 and S-CIFAR-100 under the UCL setting. All methods (with the same backbone ResNet18) are trained from scratch.

Method	Historical Model	Replay Buffer	S-CIFAR-10		S-CIFAR-100	
			acc (↑)	fg (↓)	acc (↑)	fg (↓)
LUMP+Regularization	ResNet18	256	91.23 (±0.30)	2.76 (±0.60)	82.58 (±1.30)	8.09 (±0.69)
SI+Mixup	ResNet18	256	92.90 (±0.37)	1.43 (±0.62)	80.65 (±0.62)	4.52 (±0.55)
Cassle+Mixup	ResNet18	256	91.36 (±1.33)	**0.90 (±0.74)**	79.50 (±0.82)	**1.08 (±0.20)**
RM-SimSiam (ours)	ResNet18	256	**93.07 (±0.13)**	1.36 (±0.10)	**83.26 (±0.30)**	2.73 (±0.42)

validates the effectiveness of our proposed RM strategy. (2) Our proposed RM strategy + SimSiam consistently outperforms this strategy + BarlowTwins, with about 1–2% higher accuracies on all the three benchmark datasets. This shows that the URL approach SimSiam can learn better representations as compared to BarlowTwins.

E Comparison to BYOL for UCL

Note that the well-known URL approach BYOL (Grill et al., 2020) also deploys the momentum encoder (like the hist-module in our RM-SimSiam) during model training. However, our RM-SimSiam is different from BYOL in that: (1) Because of inducing the historical and new models, the ESC loss of our RM-SimSiam can be defined over four views for knowledge distillation, which is more comprehensive than the distillation-based loss of BYOL. (2) During the rememory process of our RM-SimSiam, the knowledge of the historical model is directly transferred to the new model for mitigating the catastrophic forgetting, and the new model is further updated by backpropagation. This rememory process is otherwise ignored in BYOL.

To further the effectiveness of our RM-SimSiam, we compare it to two implementations of BYOL under the UCL setting: (1) BYOL+FINETUNE: BYOL is directly applied to UCL across all tasks; (2) BYOL+Mixup: BYOL is combined with the Mixup strategy proposed in LUMP. The comparative results in terms of average accuracy and average forgetting (over three independent runs) are shown in Table 7. It can be clearly observed that our RM-SimSiam outperforms BYOL due to the new components (i.e., our ESC loss and the rememory process) devised for UCL.

F Fair Memory Comparison

In Table 1 of the main paper, we have made direct comparison to the representative/state-of-the-art methods (without any modifications) under the UCL setting. However, these methods have different memory requirements (historical model or replay buffer), and the comparison seems somewhat unfair. To remedy this, we choose to modify the original representative/state-of-the-art methods (three selected) for fair memory comparison: (1) LUMP+Regularization: LUMP (Madaan et al., 2021) enhanced by the regularization loss for aligning the historical and new models on the replay buffer; (2) SI+Mixup: SI (Zenke et al., 2017) enhanced by the Mixup strategy proposed in LUMP; (3) Cassle+Mixup: Cassle (Fini et al., 2022) enhanced by the Mixup strategy proposed in LUMP. The comparative results in Table 8 show that our RM-SimSiam still performs the best under the same memory requirements (historical model + replay buffer).

G Details of the Mixup Strategy

As we have mentioned, there are three classic strategies introduced to mitigate the catastrophic forgetting for continual learning, including regularization-based strategy, architecture-based strategy and rehearsal-based strategy. Among them, the rehearsal-based strategy is proved to be very effec-

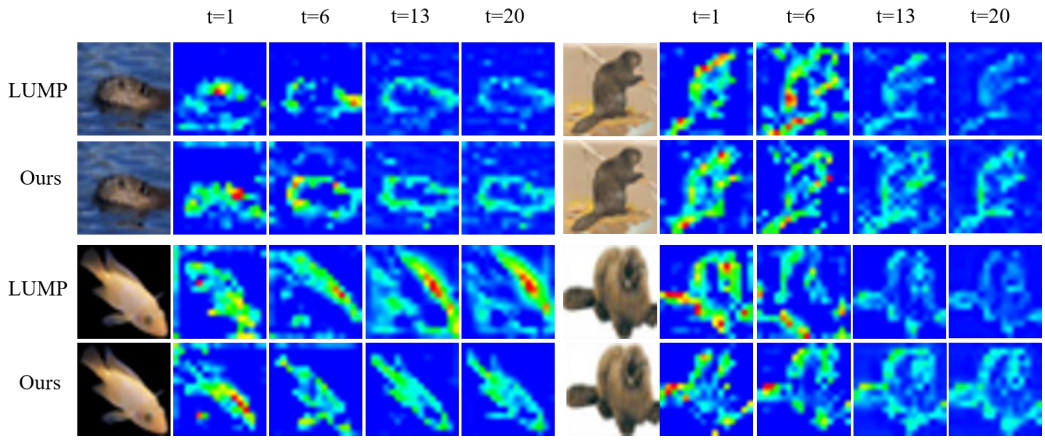

Figure 4: Visualization examples of feature maps from the second block of the backbone ResNet18 when LUMP and our RM-SimSiam are being trained sequentially on S-CIFAR-100 (but only task 1, 6, 13 and 20 are shown). The input images are randomly selected from the test set of task 1.

tive under multiple settings of continual learning (Rebuffi et al., 2017; Lopez-Paz & Ranzato, 2017; Buzzega et al., 2020; Madaan et al., 2021; Fini et al., 2022; Arani et al., 2022), such as combining the Mixup strategy (Zhang et al., 2017) with knowledge distillation (Hinton et al., 2015; Sarfraz et al., 2021) or directly deploying it. Here, we mainly introduce the Mixup strategy used in this work. Specifically, Mixup denotes an interpolation between the instances of current task and the instances of previous tasks. The instances of previous tasks are stored in a replay buffer M. Given the two original inputs x_1, x_2, and the obtained two inputs (a sample pair) from memory buffer $(x_{1,M}, x_{2,M})$, we can formulate the two interpolated inputs $\tilde{x}_1$ and $\tilde{x}_2$ as:

$$\tilde{x}_1 = \lambda \cdot x_1 + (1 - \lambda) \cdot x_{1,M}, \tag{10}$$

$$\tilde{x}_2 = \lambda \cdot x_2 + (1 - \lambda) \cdot x_{2,M}, \tag{11}$$

where λ denotes the interpolated coefficient between the two inputs. Note that λ is obtained from a beta distribution about interpolation hyperparameter α ($\lambda \in [0, 1]$).

H More Visualizations of Feature Maps

We provide more visualization examples of feature maps from (the last layer of) the second block of the backbone ResNet18 when the state-of-the-art LUMP (Madaan et al., 2021) and RM-SimSiam are being trained sequentially on task 1, 6, 13 and 20 of S-CIFAR-100 in Figure 4. And in Figure 5, we also provide the visualization examples of feature maps from (the last layer of) the second block of the backbone ResNet18 when our RM-SimSiam is being trained sequentially on task 1, 6, 13 and 20 of S-TINY-IMAGENET. From these two figures, we can observe that our RM-SimSiam is still able to learn the visual features of objects well as the number of tasks increases. This further demonstrates the effectiveness of our RM-SimSiam in continual learning.

Moreover, we also provide the visualization examples of feature maps from (the last layer of) the second block of the backbone ResNet18 when our RM-SimSiam is being trained sequentially on S-CIFAR-100 and tested on the out-of-distribution (OOD) datasets including MNIST (LeCun et al., 2010) and FMNIST (Xiao et al., 2017) in Figure 6. From the visualization of feature maps on these two OOD datasets, we can clearly observe that our RM-SimSiam is able to represent the visual features of unseen objects well even under the OOD setting. This directly demonstrates the good generalization ability of our RM-SimSiam.

I Further t-SNE Visualization Analysis

In addition to Tables 4-5 of the main paper, we provide further t-SNE visualization analysis of our RM-SimSiam on S-CIFAR-10 in Figure 7 and Figure 8 to show the impacts of two important hyperparameters α and m on the performance of our RM-SimSiam. When all tasks are learned, we exploit the features learned by the last model on each task for t-SNE visualization analysis. From

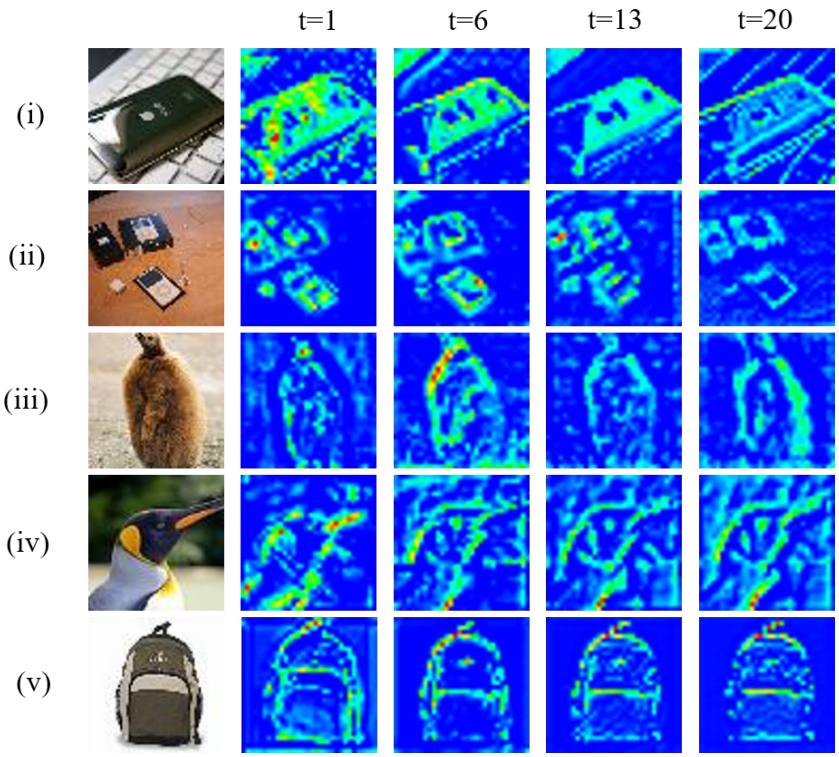

Figure 5: Visualization examples of feature maps from the second block of the backbone ResNet18 when our RM-SimSiam is being trained sequentially on S-TINY-IMAGENET (only task 1, 6, 13 and 20 are shown). The five input images are randomly selected from the test set of task 1.

Figure 6: Visualization examples of feature maps from the second block of the backbone ResNet18 when our RM-SimSiam is being trained sequentially on S-CIFAR-100 and tested on the OOD datasets, i.e., MNIST and FMNIST. The input images are randomly selected from the test set of two OOD datasets, respectively. The images in the first row are from MNIST test set, and the images in the third row are from FMNIST test set.

Figure 7, we have the following observations: (1) In the last task T_5, the only two classes are better separated when $\alpha = 0.30$; (2) In the first four tasks, the only two classes in each task are not well separated, especially on task T_2 due to the forgetting during continual model training. However, we can still find that the only two classes in each task are separated the best when $\alpha = 0.30$. That is, when α gradually increases from 0.22 to 0.34 with $m = 0.99$ fixed, our RM-SimSiam achieves the

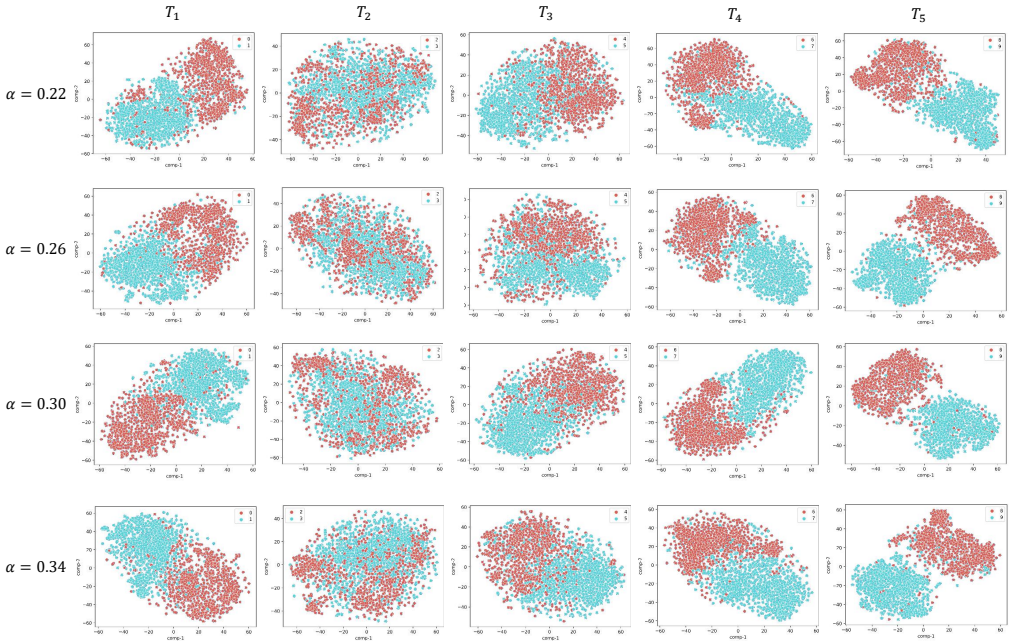

Figure 7: The t-SNE visualization analysis of our RM-SimSiam on S-CIFAR-10 when α gradually increases from 0.22 to 0.34 (but with $m = 0.99$ fixed). Our RM-SimSiam achieves the best performance with $\alpha = 0.30$.

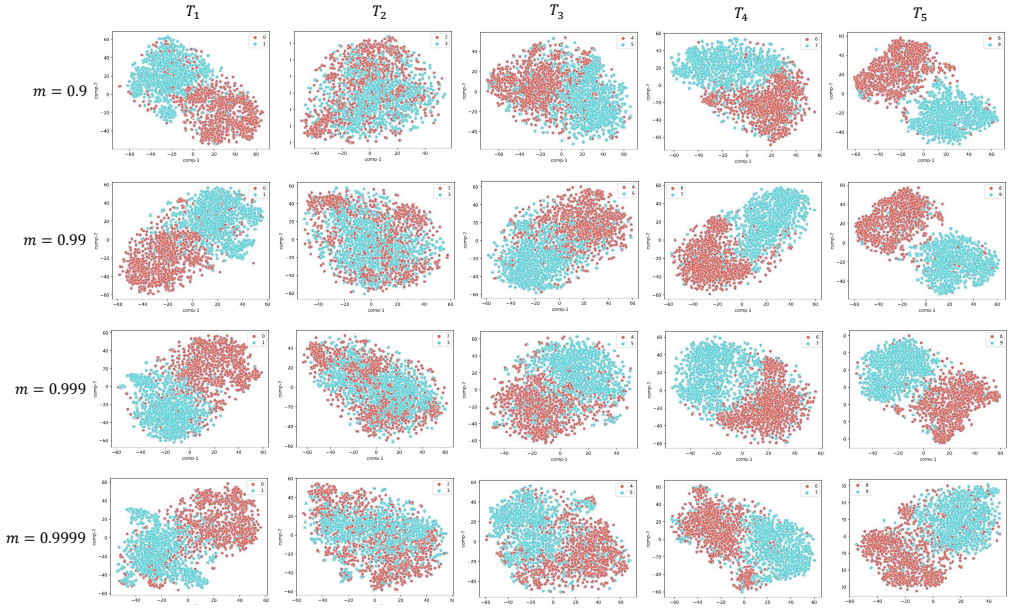

Figure 8: The t-SNE visualization analysis of our RM-SimSiam on S-CIFAR-10 when m gradually increases from 0.9 to 0.9999 (but with $\alpha = 0.30$ fixed). Our RM-SimSiam achieves the best performance with $m = 0.99$.

best performance with $\alpha = 0.30$. Similarly, as for the five tasks in Figure 8, we can see that when $m = 0.99$ (with $\alpha = 0.30$ fixed), the only two classes in each task are better separated in general, especially on the task T_1, T_4 and T_5. That is, when m gradually increases from 0.9 to 0.9999 with $\alpha = 0.30$ fixed, our RM-SimSiam achieves the best performance with $m = 0.99$.