Reduce the Difficulty of Incremental Learning With Self-Supervised Learning

LINTING GUAN AND YAN WU

1College of Electronics and Information Engineering, Tongji University, Jiading, Shanghai 201804, China
2College of Information Engineering, Zhejiang Ocean University, Lincheng, Zhoushan, Zhejiang 316004, China

Corresponding author: Yan Wu (yanwu@tongji.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant U19A2069, and in part by the General Scientific Research Projects of Zhejiang Education Department, in 2020, under Grant Y202044676.

ABSTRACT Incremental learning requires a learning model to learn new tasks without forgetting the learned tasks continuously. However, when a deep learning model learns new tasks, it will catastrophically forget tasks it has learned before. Researchers have proposed methods to alleviate catastrophic forgetting; these methods only consider extracting features related to tasks learned before, suppression to extract features for unlearned tasks. As a result, when a deep learning model learns new tasks incrementally, the model needs to learn to extract the relevant features of the newly learned task quickly; this requires a significant change in the model’s behavior of extracting features, which increases the learning difficulty. Therefore, the model is caught in the dilemma of reducing the learning rate to retain existing knowledge or increasing the learning rate to learn new knowledge quickly. We present a study aiming to alleviate this problem by introducing self-supervised learning into incremental learning methods. We believe that the task-independent self-supervised learning signal helps the learning model extract features not only effective for the current learned task but also suitable for other tasks that have not been learned. We give a detailed algorithm combining self-supervised learning signals and incremental learning methods. Extensive experiments on several different datasets show that self-supervised signal significantly improves the accuracy of most incremental learning methods without the need for additional labeled data. We found that the self-supervised learning signal works best for the replay-based incremental learning method.

INDEX TERMS Incremental learning, self-supervised learning, deep learning.

I. INTRODUCTION

The long-term goal of artificial intelligence is to build an agent that can act autonomously for a long time; which requires the agent to continuously learn new tasks to adapt to the changing environment and deal with various unknown new objects after training and deployed in the natural environment. However, the agent model based on the deep neural network will essentially forget the previously learned tasks when performing incremental learning new tasks. Specifically, when a neural network model is learning a new task, if only the samples of the new task are used to train the neural network, the recognition accuracy of the learned task will be significantly reduced. This phenomenon is called the catastrophic forgetting problem [1]. The main reason is that when the neural network model learns new tasks serially, the newly learned knowledge will overwrite the old knowledge. A trivial solution to prevent catastrophic forgetting is to keep all learned samples. When the model needs to learn a new task, let the model relearn all learned tasks simultaneously; this solution is unsustainable because storing all the learned samples and retraining the model from scratch requires massive storage space and computing resources. Since the storage space and computing resources carried by the mobile agent are minimal, it cannot meet the requirements of storing all the learned samples, so the purpose of incremental learning research is to overcome catastrophic forgetting under the condition of limited computing and storage resources, provide the agent with the ability to continue learning. In recent years, this problem has received more and more attention from researchers, and many methods have been proposed to solve the catastrophic forgetting problem [2]–[5] through various constraints. However, it does not consider that the feedback signal obtained by learning the current task will
inhibit the extraction of relevant features of the future learning task in the scenario of incremental learning. For example, when the model recognizes a dog, it needs to treat the cat in the picture as the background, not extract its relevant features. Still, when the model needs to learn new tasks incrementally, such as when recognizing cats, it also needs to learn quickly to extract features related to the new class (cat); this requires a significant change in the behavior of the model to extract features, thereby increasing the difficulty of model optimization, putting the model into the dilemma of stability/plasticity of reducing the learning rate to retain existing knowledge or increasing the learning rate to learn new knowledge [6] quickly. Figure 1 schematically shows that the model requires different responses to the same input sample in the above situation.

This paper proposes to introduce self-supervised learning to alleviate this problem. The self-supervised learning method obtains supervision signals by customizing various pseudo-labels, such as cropping, resizing, color distorting, and adding Gaussian noise to generate multiple image samples. Let the model determine whether it comes from the same picture or not [7], rotate the image to a specific angle and let the model predict the rotation angle of the image [8], and shuffle the order of the image after dividing it into blocks and let the model predict the original position [9]. Because the features extracted by the network model based on self-supervised learning have universal applicability independent of the specific subtasks of incremental learning, the neural network model trained based on self-supervised signals has achieved significant performance improvements in multiple subsequent tasks (such as image classification, target detection) [10], [11]. We integrate the supervision signals of self-supervised learning into the incremental learning process, provide constraints for the model in the incremental learning training process, and prompt the model not only to extract features that are effective for the current learned task but also extract features that have potential effects on other unlearned tasks. So that when the model learns new tasks incrementally, it can iterate more smoothly and quickly to the parameter space points that meet the new and old tasks simultaneously. Figure 2 illustrates the parameter optimization trajectory when the network model parameters are trained using different methods. After completing the learning of task one, the model is in the parameter area with a lower error rate for task one. When the model continues to train using the samples in task two, the model parameters will be iterated to a region with a lower error rate for task two, such as the yellow area shown by the red arrow. Still, this area has a higher error rate for task one, which means the model has disastrously forgotten the knowledge learned from task one. When using the incremental learning method, the model will optimize to the yellow parameter area where both task one and task two perform well, as shown by the yellow arrow, this area has a lower error rate for both task one and task two, but this area is far from the optimal parameter area (such as the green area where the circle is located) of the unlearned task (e.g., task three). Incremental learning combined with self-supervised learning can optimize the parameters of the model to balance between the current learned tasks (such as task one and task two) and the parameter space point closer to the optimal parameter area of the unlearned task (e.g., task three), such as the blue area shown by the blue arrow in the picture. By combining different databases and incremental learning methods, we have verified that fusing self-supervised and supervised learning signals can significantly improve the accuracy of the class incremental learning (class-IL) method without additional labeling data or storing more samples. In particular, self-supervised learning has a significant effect on sample replay-based class-IL.

II. RELATED WORK
A. INCREMENTAL LEARNING
Early scholars encountered the catastrophic forgetting problem in the 1980s [1] when studying incremental learning based on neural networks. Kolen et al. found that the main reason is that there are cliff-like changes in the parameter space; small changes in model parameters caused by incremental learning may cause drastic changes in model
output [12]. Some early related work includes the introduction of multiple network architectures [13], increasing the independence of features [14], and replaying samples, etc. [15], [16]. However, due to the limitations of computing resources, these works only used few samples and shallow neural network architecture. With the popularity of deep neural networks in recent years, more and more works have tried to solve the catastrophic forgetting problem in incremental learning. These incremental learning research works can be roughly divided into replay-based methods [17]–[19], regularization-based methods [2], [5], [20], and parameter isolation methods [3], [4].

Early scholars encountered the catastrophic forgetting problem in the 1980s [1] when studying incremental learning based on neural networks. Kolen et al. found that the main reason is that there are cliff-like changes in the parameter space; small changes in model parameters caused by incremental learning may cause drastic changes in model output [12]. Some early related work includes the introduction of multiple network architectures [13], increasing the independence of features [14], and replaying samples, etc. [15], [16]. However, due to the limitations of computing resources, these works only used few samples and shallow neural network architecture. With the popularity of deep neural networks in recent years, more and more works have tried to solve the catastrophic forgetting problem in incremental learning. These incremental learning works can be roughly divided into replay-based methods [17]–[19], regularization-based methods [2], [5], [20], and parameter isolation methods [3], [4].

The method based on model parameter constraints reduces the forgetting of learned tasks by introducing additional constraints to the loss function. This type of method does not need to store the trained samples, thereby reducing memory. According to the different constraints, such methods can be divided into data-based constraint methods and a priori constraint-based methods. When the model starts training a new task, LwF [25] uses the output of the new sample in the old model as a soft label to constrain the update of model parameters, to reduce the degree of forgetting of the learned task in the model. Elastic Weight Consolidation (EWC) [5], Memory Aware Synapses (MAS) [26], and Intelligent Synapses (SI) [20] calculate the importance of model parameters to the learned task and using it as a constraint condition for updating the parameters, reduce the update range of essential parameters when learning new tasks, thereby reducing the degree of forgetting of the learned tasks by the model. This type of method requires additional space to save the importance of the model parameters in the learned task.

Parameter isolation-based methods such as Progressive Neural Networks (PNN) [3] freeze the network parameters of all old tasks and learn new tasks by building a new network each time. To retain the previously learned knowledge, the model will simultaneously input the data of the new task into all old frozen networks and merge the output of each network to prevent the forgetting of the learned tasks when learning new tasks; Dynamically Expandable Networks (DEN) [4] participate in the training of new tasks by dynamically selecting some of the parameter nodes of the existing network, and expand the parameters of the network when needed to reduce the forgetting of learned tasks; Expert Gate (EG) [27] select and copy an old model with the highest similarity from the old task for the new task through a self-encoding gate for training the new task. The Deep Adaptation Networks (DAN) [28] frozen the learned filters of the old tasks, but allows for learning new filters that are linear combinations of existing filters from new tasks. Such methods need to continuously expand the capacity of the model when learning new tasks.

B. SELF-SUPERVISED LEARNING

Traditional supervised deep learning methods rely heavily on the amount of labeled data. However, data labeling is very time-consuming, laborious, and easy to introduce noise, which significantly limits the development of deep learning. The self-supervised learning method uses the information of the data itself to construct pseudo-labels to obtain supervised signals and gets rid of the dependence on artificially labeled data. The feature representation extracted based on self-supervised learning has improved performance in multiple subsequent tasks such as image classification, object detection, and semantic segmentation. Scholars have proposed a variety of methods to construct pseudo-labels from raw data, such as using cropping, resize and color distortion methods to generate multiple samples from the same image and train the model to judge where a pair of images are from the same image [7]. Gidaris et al. [8] proposed to rotate the training images with random angles and train the model to predict the rotation angle. Doersch et al. [9] utilizing the order of the image after dividing it into blocks and train the model to predict the original position. Qian et al. [29] proposed to train the model to judge whether the order of the sample frames in the video is correct.

However, there are few attempts to combine self-supervised learning and incremental learning. We believe that self-supervised learning signals complement specific incremental learning tasks, so it helps to improve the generalization performance of the features extracted from the model and makes it easier to learn incrementally. The most related work with ours is [30], this work only discussed the effect of combining self-supervised signals and the OWM algorithm [31]. However, we discussed the effect of combining self-supervised learning with the current main different types of incremental learning methods, and we have done many experiments on CIFAR-10 and Tiny ImageNet datasets and found that self-supervised learning has a better effect on alleviating catastrophic forgetting in incremental learning methods based on sample replay than based on parameter isolation. In further experiments, we found that the self-supervised learning signal can amplify the cached samples to alleviate catastrophic forgetting as the cached samples increase.
Experiments in section IV show that incremental learning based on the fusion of self-supervised learning signals and sample replay can achieve the current best incremental learning performance with only a slight increase in storage and calculation costs.

III. METHODOLOGY

A. FORMULATION

In this section, we formally discuss the incremental learning problem. An incremental learning problem can be defined as letting the model \( M \) with the parameter \( \theta \) learn \( T \) tasks sequentially, the training dataset \( D_t \) of each task \( t \in \{1, \ldots, T\} \) consists of \( N_t \) training samples conforming to independent and identical distribution \( \{x_{t,i}, y_{t,i}\}_{i=1}^{N_t} \), the label sets of different tasks \( y_i \) and \( y_j \) are orthogonal to each other:

\[
y_i \cap y_j = \emptyset \quad \forall i, j \in T \land i \neq j
\]

The goal of incremental learning is to optimize the parameters \( \theta \) from the training samples to minimize the sum of the model’s losses in all \( T \) tasks:

\[
\arg \min_{\theta} \sum_{t=1}^{T} L_t \quad \text{where} \quad L_t = E_{(x,y) \sim D_t}[\ell_{ce}(y, f_\theta(x))] \tag{1}
\]

The function \( \ell_{ce} \) is the cross-entropy loss. Since incremental learning assumes that the model learns each task sequentially, the model can only be optimized based on the first \( n \) tasks currently learned each time, where \( n \leq T \), so the model minimizes the following objective function each time:

\[
\arg \min_{\theta} \sum_{t=1}^{n} L_t \quad \text{where} \quad L_t = E_{(x,y) \sim D_t}[\ell_{ce}(y, f_\theta(x))] \tag{2}
\]

The optimal solution of the above objective function is to find the parameter region with the lowest error rate for the first \( n \) tasks. The problem with this objective function is that its optimization direction does not consider the parameter area that has a lower error rate for the task to be learned in the future, so the optimal parameters obtained are easy to fall in the area far away from the optimal parameters of the unlearned task (take the \( n + 1 \)th as an example), this makes it difficult for the model to update its parameters to a region with a lower error rate for the first \( n + 1 \) tasks when it continues to learn the \( n + 1 \)th task.

We add self-supervised learning items to the incremental learning objective function, let the model balance the loss of incremental tasks \( \mathcal{L} \) and the loss of self-supervised tasks \( \mathcal{L}_{ssl} \) during optimization. The optimized objective function is as follows:

\[
\arg \min_{\theta} \sum_{t=1}^{n} \mathcal{L}_t + \alpha \mathcal{L}_{ssl} \quad \text{where} \quad \\
\mathcal{L}_t = E_{(x,y) \sim D_t}[\ell_{ce}(y, f_\theta(x))], \\
\mathcal{L}_{ssl} = E_{(x,\hat{y}) \sim D_t}[\ell_{ce}(\hat{y}, f_\theta(f_j(x)))] \tag{3}
\]

where \( \mathcal{L}_{ssl} \) is a self-supervised item, \( f_j \) transforms the input image data, \( \hat{y} \) is the pseudo label generated after the image is transformed, \( \alpha \) is a hyperparameter that balances incremental learning loss and self-supervised learning loss. Adding \( \mathcal{L}_{ssl} \) to the loss function allows the model not only to consider the label of the current task in the incremental learning process but also maintain a particular perception of other unknown objects so that the model can adapt to recognize new classes of objects faster when learning new tasks. For example, when learning to recognize a new class of objects (such as dogs), the model needs to suppress the response to unlearned objects (such as cats) that may appear in the sample without a self-supervised learning signal. After adding the self-supervised learning signal to the optimized objective function, the model needs to extract semantic information of unknown objects for self-supervised learning tasks, such as judging the orientation of unlearned objects or the position of various body parts. The self-supervised signal requires the model to retain the perception of unknown objects in the feature map. Therefore, the model will learn how to classify them when incrementally learns new classes of objects correctly.

Due to storage space limitations, the agent cannot save samples of all learning tasks in its life cycle. Incremental learning assumes that the training data for different tasks arrive in time sequence and the data of the \( n - 1 \)th task is not retained when training the \( n \)th new task, so the formula 3 cannot be calculated due to missing data in \( D_1 \ldots D_{n-1} \). Different incremental learning methods have different strategies to compensate for the missing data of \( D_1 \ldots D_{n-1} \), such as using sample replay, model parameter constraints, and parameter isolation. We use \( \mathcal{L}_{inc} \) to represent the constraints added by incremental learning, and the corresponding optimization objective function is changed to the following form:

\[
\arg \min_{\theta} \mathcal{L}_n + \alpha \mathcal{L}_{ssl} + \beta \mathcal{L}_{inc} \quad \text{where} \quad \\
\mathcal{L}_n = E_{(x,y) \sim D_n}[\ell_{ce}(y, f_\theta(x))], \\
\mathcal{L}_{ssl} = E_{(x,\hat{y}) \sim D_n}[\ell_{ce}(\hat{y}, f_\theta(f_j(x)))] \tag{4}
\]

Different incremental learning algorithms have different \( \mathcal{L}_{inc} \) formulas. For example, the constraints of regularization based methods such as EWC [5], SI [20], LwF [25] are as follows:

\[
\mathcal{L}_{EWC} = \sum_i F_i(\theta_i - \theta^*_i)^2 \\
\mathcal{L}_{SI} = \sum_i \Omega_i(\theta_i - \theta^*_i)^2 \\
\mathcal{L}_{LwF} = - \sum_j y_o^i \log \hat{y}_o^i \tag{5}
\]

where \( \theta^* \) is the model parameter after learning the previous task, \( F \) and \( \Omega \) record the importance of network weights in different forms, \( Y_o \) and \( \hat{Y}_o \) record the network output of the previous and current models on the data of current task respectively.

Replay based methods, such as ER [16], [18], Dark Experience(DER) [21], sample part of the training task data to the cache \( \mathcal{M} \), and use the data in the cache instead of \( D_1 \ldots D_{n-1} \).
to calculate the loss of the learned task, the corresponding loss function is as follows:

\[
\mathcal{L}_{\text{ER}} = E_{(x,y) \sim \mathcal{M}}[\ell_{ce}(y, f_0(x))]
\]

\[
\mathcal{L}_{\text{DER}} = E_{(x,z) \sim \mathcal{M}}[\|z - h_0(x)\|^2_2]
\]

(6)

where \(z\) is the cache of extracting features from the data \(x\) before the model is updated, and \(h_0(x)\) is the feature extracted from the data \(x\) by the current model.

We can flexibly add the self-supervised learning strategy to the replay-based methods and regularization-based methods. We used predicted image rotation [8] as the implementation method of self-supervised learning signal. In addition, for the replay-based method, we can calculate the self-supervised learning signal of self-supervised learning. In addition, for the replay-based methods and regularization-based methods.

Algorithm 1 describes the calculation process of combining self-supervised learning and DER in detail. The corresponding loss function is modified as follows:

\[
\mathcal{L}_{\text{ssl}} = E_{(x) \sim D_{\text{ssl}}}[-\frac{1}{K} \sum_{y=1}^{K} \log(f_{\theta}^y(g(x|y)))]
\]

where \(g(x|n) = \text{Rot}(x, n \times 90), D_{\text{ssl}} \in \{\mathcal{M}, \mathcal{D}_n, \mathcal{M} \cup \mathcal{D}_n\}\) (7)

where the \(\text{Rot}(:m)\) function rotates the image by \(m\) degrees, and the function \(f_{\theta}^y(.)\) predicts the probability that the model result is \(\hat{y}\), refer to [8] we set \(K\) to 4.

B. ALGORITHM DESCRIPTION

Algorithm 1 describes the calculation process of combining self-supervised learning and DER in detail. Where function \(\text{reservoir}(\mathcal{M}, (x, z))\) implements the reservoir sampling algorithm [32] to ensure that the samples in different tasks have the same probability of being stored in the buffer.

IV. EVALUATION METRICS

The literature [33] divides incremental learning into task incremental learning (task-IL), class-IL, and domain incremental learning (domain-IL). They all divide samples into multiple different tasks, and each task contains multiple different classes. The model learns each task and corresponding classes incrementally. The main difference between task-IL and class-IL is that task-IL provides information about which task the test sample belongs to during the test period, while class-IL does not provide this information, so class-IL needs to distinguish test samples from all learned classes, which is most demanding among the three incremental learning settings, while task-IL only needs to distinguish test samples belonging to the specific classes of the given task, because each task contains fewer numbers of classes, task-IL is the easiest one among the three types of incremental learning. Domain-IL assumes that the class architectures in all tasks are the same. Still, domain switching occurs in different tasks, so domain-IL does not provide task information in the test and only needs to predict that the sample belongs to a particular class within a single task, so it is more challenging than task-IL and simpler than class-IL. This section will experiment on the effect of self-supervised learning on class-IL and task-IL. Our experimental environment settings are consistent with [21], [33], [34]

A. EVALUATION PROTOCOL

Based on the universality, diversity, and complexity of the sample, we have selected the relatively simple CIFAR-10 [35], the more complex Tiny ImageNet [36] dataset, and the traffic sign dataset GTSRB [37] to comprehensively test the effectiveness of the algorithm, all results are averaged across five runs.

1) CIFAR-10

CIFAR-10 consists of 60,000 32 × 32 color pictures. These pictures belong to 10 common object classes. Each class has 6,000 pictures. The entire dataset has 50,000 training pictures and 10,000 test pictures. We divided the dataset into five tasks according to the setting of incremental learning, and each task contains two categories.
2) TINY ImageNet
They were composed of 120,000 64 × 64 color pictures, including 200 types of common objects. Each class has 500 training pictures, 50 validation pictures, and 50 test pictures. We divide Tiny ImageNet into ten tasks according to the setting of incremental learning, and each task contains 20 classes. Examples of pictures randomly sampled from the dataset are shown in Figure 3.

3) GTSRB
GTSRB is composed of more than 50,000 color pictures, including 43 classes of traffic signs. We divide GTSRB into 43 tasks according to the setting of incremental learning, and each task contains 1 class. Examples of pictures randomly sampled from the dataset are shown in Figure 4.

4) NETWORK ARCHITECTURE
Refer to [19] and [21], we use the unpretrained ResNet-18 [22] as the network architecture, the details of ResNet-18 are shown in Table 1, where \([k_w \times k_h, c_n]\) is a residual block, \(k_w\) and \(k_h\) mean kernel width and height, \(c_n\) means channel number, \([k_w \times k_h, c_n] \times r_n\) means residual block repeat \(r_n\) times, please refer to [22] for more details about the residual block.

5) DATA ENHANCEMENT STRATEGY
We enhance the training data with random cropping and horizontal flipping.

6) HYPERPARAMETER SELECTION
Refer to the hyperparameter settings of [21], we set the learning rate \(lr\) to 0.03, the batch data size to 32, and the number of data iterations for each task to 50. The hyperparameters of the self-supervised part are uniformly set to 0.05. The network training is all using Stochastic Gradient Descent(SGD) Optimizer.

B. EXPERIMENTAL RESULTS
To experimentally verify the effect of self-supervised learning on incremental learning, we selected Regularization-based Methods (LwF, EWC, SI) and Replay-based Methods (ER, DER, DERPP). We have not verified the effect of the parameter isolation-based incremental learning methods because the parameter isolation-based incremental learning method does not effectively prevent the catastrophic forgetting of the model based on the experimental results of [21]. We did detailed experiments on class-IL and task-IL. We set the number of batch samples to 32 in all experiments. For the regularization-based methods, we add the self-supervision directly to the new task samples; for the replay-based methods, we add the self-supervised signal to the cached samples and fix the number of cached samples to 200; all results are the average precision and variance of 5 experiments.

Table 2 and Table 5 show the experimental results in the CIFAR database and Tiny ImageNet respectively. As seen in the class-IL, adding self-supervised learning to different incremental learning methods has a general improvement effect on classification accuracy. In particular, self-supervised learning can significantly improve the classification accuracy of incremental learning methods based on sample replay. As we can see in Table 2, the accuracy of DER’s class-IL in CIFAR-10 increased from 75.47% to 77.27%, the accuracy of task-IL has also increased from 92.88% to 93.78%, Self-supervised learning also improves ER and DERPP. Therefore, adding self-supervised learning has obvious help for
the improvement of class-IL. While, self-supervised signal has a slight negative effect on the accuracy of LwF method, we believe that this is because the LwF method needs to exactly fit the output of the previous tasks, so the model is less robust to the introduction of new supervision signals. Table 4 shows the p-value of the experimental results from the statistical aspect, as we can see from the result that the self-supervised signal has very high confidence in the improvement effect of the replay-based method. Figure 5 and Figure 7 shows the impact of various incremental learning algorithms on class-IL by self-supervised learning signals in the CIFAR-10 and Tiny ImageNet datasets, respectively.

Since we set the GTSRB data set to contain only one class per task, the model needs to perform incremental learning on a task sequence with a total of 43 tasks. After preliminary experiments on the regular-based methods, we found that this method cannot effectively preserve knowledge during such a long incremental learning sequence of tasks. For example, the LWF based method only has an average accuracy of 7.8%, but the replay-based methods sample works well. Therefore, we focused on experiments on the replay-based method (ER, DER, DERPP). In the experiment of the GTSRB dataset, we set the number of batch samples to 32 and fixed the number of cached samples to 1000. We tested the effects of category incremental learning and task-IL. We found that the results of all task-IL reached 100%. Hence, the table 3 only shows the experimental results of category incremental learning in the GTSRB dataset.

As we can see in the class-IL, adding self-supervised learning to different replay-based incremental learning meth-

Table 2. Experimental results of whether to add self-supervised learning to different incremental learning algorithms in the CIFAR-10 dataset.

| Dataset | Method | Class-IL | Task-IL |
|---------|--------|----------|---------|
| CIFAR-10 | LWF [25] | 43.91±0.18 | 74.62±0.98 |
| LWF+SSL | 43.97±0.03 ↑ | 75.84±0.94 ↑ | 77.30±0.80 ↓ |
| Si [20] | 43.89±0.21 | 74.38±0.21 | 77.30±0.80 |
| Si-SSL | 44.12±0.19 ↑ | 75.50±0.45 ↓ | 77.30±0.80 |
| EWC [5] | 39.93±0.10 ↑ | 67.72±4.59 | 77.30±0.80 |
| EWC+SSL | 42.76±0.52 ↑ | 73.31±1.45 ↑ | 77.30±0.80 |
| ER [18] | 65.93±0.05 | 92.91±0.12 | 77.30±0.80 |
| ER+SSL | 68.23±0.19 ↑ | 93.01±0.13 ↑ | 77.30±0.80 |
| DER [21] | 75.47±0.16 | 92.88±2.47↑ | 77.30±0.80 |
| DER+SSL | 77.27±1.69↑ | 93.78±0.71↑ | 77.30±0.80 |
| DERPP [21] | 77.79±0.46↑ | 94.17±0.09↑ | 77.30±0.80 |
| DERPP+SSL | 78.9±0.26↑ | 93.91±0.19↓ | 77.30±0.80 |

Table 3. Experimental results of whether to add self-supervised learning to replay based incremental learning algorithms in the GTSRB dataset.

| Dataset | Method | Class-IL |
|---------|--------|----------|
| GTSRB   | ER [18] | 94.14±0.99 |
| ER+SSL  | 96.12±0.35 ↑ | 77.30±0.80 |
| DER [21] | 88.27±4.02 | 77.30±0.80 |
| DER+SSL | 91.68±0.77 ↑ | 77.30±0.80 |
| DERPP [21] | 95.56±0.26 | 77.30±0.80 |
| DERPP+SSL | 95.86±0.26↑ | 77.30±0.80 |

Table 4. The confidence of self-supervised signal on the accuracy improvement of replay based methods in the CIFAR-10 dataset.

| Dataset | Method | Accuracy |
|---------|--------|----------|
| CIFAR-10 | ER      | 74.62±0.98 |
| ER+SSL  | 75.84±0.94 ↑ | 77.30±0.80 ↓ |
| DERPP+SSL | 93.78±0.71↑ | 77.30±0.80 |

Figure 5. The influence of self-supervised learning on the class-IL accuracy of different incremental learning methods in the CIFAR-10 dataset.

Figure 6. The influence of self-supervised learning on the task-IL accuracy of different incremental learning methods in the CIFAR-10 dataset.
TABLE 5. The impact of Whether to add a self-supervised learning signal on the average accuracy (%) of different incremental learning algorithms in Tiny ImageNet dataset.

| Database   | Method | Class-IL | Task-IL |
|------------|--------|----------|---------|
| Tiny ImageNet | LWF [23] | 23.84 ± 1.32 | 29.02 ± 0.06 |
|            | LWF+SSL | 22.73 ± 0.35 | 31.45 ± 0.73 |
|            | SI [20]  | 20.51 ± 0.09 | 34.08 ± 0.25 |
|            | SI+SSL  | 20.88 ± 0.02 | 33.14 ± 0.07 |
|            | EWC [5]  | 20.95 ± 0.06 | 30.34 ± 0.19 |
|            | EWC+SSL | 21.45 ± 0.06 | 30.66 ± 0.23 |
|            | ER [18]  | 23.74      | 51.24    |
|            | ER+SSL  | 24.22 ± 0.01 | 51.49 ± 0.06 |
|            | DER [21] | 29.48 ± 0.11 | 54.91 ± 0.17 |
|            | DER+SSL | 30.02 ± 0.85 | 53.3 ± 0.95 |
|            | DERPP [21]| 28.07 ± 1.49 | 51.97 ± 6.2 |
|            | DERPP+SSL | 29.07 ± 0.09 | 52.45 ± 0.67 |

FIGURE 7. Influence of self-supervised learning on the class-IL accuracy of different incremental learning methods in the Tiny ImageNet dataset.

FIGURE 8. Influence of self-supervised learning on the task-IL accuracy of different incremental learning methods in the Tiny ImageNet dataset.

or and cached samples. This section examines the impact of self-supervised signals generated from new task samples, cached samples, and new task samples + cached samples on incremental learning. We ran five trials on CIFAR-10 and Tiny ImageNet using the self-supervised + DER (RDER) method. The average accuracy and variance are shown in the Table 6, as we can see from the results that the self-supervised signal only acts on the cache to improve the accuracy of incremental learning more than it acts on the new samples or both of the new samples and cache samples. We can also find that the more cached samples, the better effect of self-supervised learning on the accuracy of incremental learning. We can conclude from the results that the self-supervised learning signal has a significant effect on improving the accuracy of replay-based incremental learning algorithms, and the key

is to apply the self-supervised learning signal to the cached sample.

2) THE EFFECT OF CACHE SIZE ON SELF-SUPERVISED SIGNAL

This section investigates the effect of buffer size on self-supervised signals in the replay-based incremental learning algorithm. We set the cache size to 200, 500, and 1000 in the experiment and repeat experiments five times on the CIFAR-10 and Tiny ImageNet datasets; the average results and its variance of experiments are shown in Table 7 and Table 8 as we can see from the results that the accuracy of class-IL is significantly improved as the cache size increases. The following is an explanation and analysis of the experimental results. When the cache size is set to 200, the average accuracy of class-IL has been improved by 2.06% and 0.76% in the CIFAR-10 and Tiny ImageNet datasets, respectively. When the cache size is increased to 500, the average accuracy of class-IL is improved by 2.86% and 1.73% in the CIFAR-10 and Tiny ImageNet datasets, respectively; this shows that the effect of the self-supervised signal on the incremental learning model increases significantly with the increase of the cache size. As the cache size increased to 1000, the average accuracy of class-IL is improved by 5.13% and 1.86% in the CIFAR-10 and Tiny ImageNet datasets, respectively; this shows that the accuracy of the self-supervised signal on the incremental learning model continues to improve with the continuous increase of the cache size. We believe this is because the self-supervised signal can more fully dig out the potential information in the cached samples, thereby better alleviating the model’s forgetting of learned knowledge in the incremental learning process. We also find that self-supervised signals do not significantly help the improvement of task-IL.

TABLE 6. The effects of self-supervised learning signals acting on new task samples, cached samples, and both new and cached samples.

| Sample domain | cache size | CIFAR-10 | Tiny ImageNet |
|---------------|------------|----------|---------------|
| new           | 200        | 71.39 ± 2.19 | 27.3 ± 0.03 |
| cache         | 200        | 77.27 ± 1.69 | 30.02 ± 0.85 |
| new+cache     | 200        | 73.94 ± 0.04 | 29.74 ± 0.04 |
| new           | 500        | 71.30 ± 0.28 | 32.3 ± 0.2   |
| cache         | 500        | 81.65 ± 0.04 | 37.81 ± 0.12 |
| new+cache     | 500        | 79.75 ± 0.03 | 36.71 ± 0.52 |
| new           | 1000       | 73.17 ± 2.18 | 39.77 ± 0.55 |
| cache         | 1000       | 84.48 ± 0.40 | 43.69 ± 0.20 |
| new+cache     | 1000       | 82.36 ± 4.18 | 42.11 ± 0.38 |

TABLE 7. Average accuracy and variance of DER and DER+SSL methods in the CIFAR-10 dataset.

| Cache Size | Method | Class-IL | Task-IL |
|------------|--------|----------|---------|
| 200        | DER    | 75.21 ± 0.16 | 93.67 ± 0.13 |
| 200        | DER+SSL| 77.27 ± 1.69 | 93.78 ± 0.71 |
| 500        | DER    | 78.89 ± 2.53 | 94.77 ± 0.07 |
| 500        | DER+SSL| 81.65 ± 0.04 | 94.41 ± 0.08 |
| 1000       | DER    | 79.35 ± 0.54 | 95.33 ± 0.11 |
| 1000       | DER+SSL| 84.48 ± 0.40 | 94.94 ± 0.23 |
VI. CONCLUSION

Incremental learning requires continuous learning of new tasks and overcoming the network’s catastrophic forgetting of learned tasks. Taking advantage of the fact that self-supervised learning and incremental learning have nothing to do with specific tasks, we add self-supervised learning to the incremental learning method to smooth the enormous changes in feature extraction behavior due to model learning in different tasks reducing the difficulty of learning. We added self-supervised learning signals to six different incremental learning algorithms. We did many experiments on CIFAR-10 and Tiny ImageNet to test the influence of self-supervised learning on incremental learning. Experimental results show that self-supervised learning can significantly improve the accuracy of class-IL algorithms, especially in the replay-based incremental learning method; its effect is significantly improved as the number of cached samples increases. In addition, we also found in experiments that adding self-supervised learning to the cached samples is more effective than adding to the new task samples or adding to both the new task samples and cached samples simultaneously; this shows that adding self-supervised learning can more effectively retain the knowledge of learned tasks through cached samples. This feature allows us to use a lower computational cost to significantly improve the performance of replay-based incremental learning methods. Recently, contrastive self-supervised learning has made significant progress. However, contrastive self-supervised learning is more challenging to converge; it requires more negative samples than pseudo-label-based methods, so designing a suitable contrastive self-supervised method to make it suitable for incremental learning is what we plan to do shortly. In the long term, we may also try to introduce methods such as the automatic discovery of high-order patterns [38] to help incremental learning methods better alleviate catastrophic forgetting.

REFERENCES

[1] M. McCloskey and N. J. Cohen, “Catastrophic interference in connectionist networks: The sequential learning problem,” Psychol. Learn. Motivat., vol. 24, pp. 109–165, Dec. 1989.

[2] J. Schwarz, W. Czarnecki, J. Luketina, A. Grabska-Barwinska, Y. W. Teh, R. Pascanu, and R. Hadsell, “Progress & compress: A scalable framework for continual learning,” in Proc. Int. Conf. Mach. Learn., 2018, pp. 4528–4537.

[3] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, “Progressive neural networks,” 2016, arXiv:1606.04671. [Online]. Available: http://arxiv.org/abs/1606.04671

[4] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, “Lifelong learning with dynamically expandable networks,” 2017, arXiv:1708.01547. [Online]. Available: http://arxiv.org/abs/1708.01547

[5] K. James, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, and D. Hassabis, “Overcoming catastrophic forgetting in neural networks,” Proc. Nat. Acad. Sci. USA, vol. 114, no. 13, pp. 3521–3526, Mar. 2017.

[6] W. C. Abraham and A. Robins, “Memory retention—the synaptic stability versus plasticity dilemma,” Trends Neurosci., vol. 28, no. 2, pp. 73–78, Feb. 2005.

[7] O. Henaff, “Data-efficient image recognition with contrastive predictive coding,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 4182–4192.

[8] S. Gidaris, P. Singh, and N. Komodakis, “Unsupervised representation learning by predicting image rotations,” arXiv:1803.07728. [Online]. Available: http://arxiv.org/abs/1803.07728

[9] C. Doersch, A. Gupta, and A. A. Efros, “Unsupervised visual representation learning by context prediction,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 1422–1430.

[10] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Baranerjee, and F. Makedon, “A survey on contrastive self-supervised learning,” Technologies, vol. 9, no. 1, p. 202, Dec. 2020.

[11] L. Jing and Y. Tian, “Self-supervised visual feature learning with deep neural networks: A survey,” IEEE Trans. Pattern Anal. Mach. Intell., early access, May 4, 2020, doi: 10.1109/TPAMI.2020.2992593.

[12] J. B. Pollack, “Backpropagation is sensitive to initial conditions,” Complex Syst., vol. 4, pp. 269–280, 1990.

[13] B. Ans and S. Roussel, “Avoiding catastrophic forgetting by coupling two reverberating neural networks,” Comp. Rendus de l’Académie des Sci. Ser. III Sci. de la Vie, vol. 320, no. 12, pp. 989–997, Dec. 1997.

[14] R. M. French, “Semi-distributed representations and catastrophic forgetting in connectionist networks,” Connection Sci., vol. 4, nos. 3–4, pp. 365–377, Jan. 1992.

[15] D. L. Silver and R. E. Mercer, “The task rehearsal method of life-long learning: Overcoming impoverished data,” in Proc. Conf. Can. Soc. Comput. Stud. Intell. Berlin, Germany: Springer, 2002, pp. 90–101.

[16] R. B. Catell, “Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions,” Psychol. Rev., vol. 97, no. 2, p. 285, 1990.

[17] A. Robins, “Catastrophic forgetting, rehearsal and pseudorehearsal,” Connection Sci., vol. 7, no. 2, pp. 123–146, 1995.

[18] M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, and G. Tesauro, “Learning to learn without forgetting by maximizing transfer and minimizing interference,” in Proc. Int. Conf. Learn. Represent., 2019, pp. 1–31. [Online]. Available: https://openreview.net/pdf?id=Bl7TShAct7 and https://icl.cc/Conferences/2019/Schedule/?type=Poster

[19] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, “ICarl: Incremental classifier and representation learning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 2001–2010.

[20] F. Zenke, B. Poole, and S. Ganguli, “Continual learning through synaptic intelligence,” in Proc. Int. Conf. Mach. Learn., Jul. 2017, pp. 3987–3995.

[21] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara, “Dark experience for general continual learning: A strong, simple baseline,” 2020, arXiv:2004.07211. [Online]. Available: http://arxiv.org/abs/2004.07211

[22] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

[23] D. Lopez-Paz and M. Ranzato, “Gradient episodic memory for continual learning,” in Proc. Int. Conf. Mach. Learn., 2017, pp. 381–389.

[24] K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, and D. Hassabis, “Dynamically expandable networks,” 2017, arXiv:1706.08840. [Online]. Available: http://arxiv.org/abs/1706.08840

[25] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, “Efficient lifelong learning with A-GEM,” 2018, arXiv:1812.00420. [Online]. Available: http://arxiv.org/abs/1812.00420

[26] Z. Li and D. Hoiem, “Learning without forgetting.” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 12, pp. 2935–2947, Dec. 2018.

[27] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, “Memory aware synapses: Learning what (not) to forget,” in Proc. Conf. Eur. Conf. Comput. Vis., Sep. 2018, pp. 139–154.

[28] R. Aljundi, P. Chakravarty, and T. Tuytelaars, “Expert gate: Lifelong learning with a network of experts,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 3366–3375.

TABLE 8. Average accuracy and variance of DER and DER+SSL methods in the Tiny ImageNet dataset.

| Cache Size | Method     | Class-IL | Task-IL     |
|------------|------------|----------|-------------|
| 200        | DER        | 0.26±0.11| 0.24±0.17   |
| 200        | DER+SSL    | 0.30±0.05| 0.33±0.05   |
| 500 vol. 24 | DER+SSL    | 0.36±0.12| 0.15±0.06   |
| 500        | DER        | 0.37±0.01| 0.65±0.06   |
| 1000       | DER        | 0.41±0.03| 0.65±0.06   |
| 1000       | DER+SSL    | 0.43±0.08| 0.65±0.06   |

L. Guan, Y. Wu: Reduce Difficulty of Incremental Learning With Self-Supervised Learning
L. Guan, Y. Wu: Reduce Difficulty of Incremental Learning With Self-Supervised Learning

[28] A. Rosenfeld and J. K. Tsotsos, “Incremental learning through deep adaptation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 3, pp. 651–663, Mar. 2020.

[29] R. Qian, T. Meng, B. Gong, M.-H. Yang, H. Wang, S. Belongie, and Y. Cui, “Spatiotemporal contrastive video representation learning,” 2020, arXiv:2008.03800. [Online]. Available: http://arxiv.org/abs/2008.03800

[30] S. Zhang, G. Shen, J. Huang, and Z.-H. Deng, “Self-supervised learning aided class-incremental lifelong learning,” 2020, arXiv:2006.05882. [Online]. Available: http://arxiv.org/abs/2006.05882

[31] G. Zeng, Y. Chen, B. Cui, and S. Yu, “Continual learning of context-dependent processing in neural networks,” Nature Mach. Intell., vol. 1, no. 8, pp. 364–372, Aug. 2019.

[32] J. S. Vitter, “Random sampling with a reservoir,” ACM Trans. Math. Softw., vol. 11, no. 1, pp. 37–57, 1985.

[33] G. M. van de Ven and A. S. Tolias, “Three scenarios for continual learning,” 2019, arXiv:1904.07734. [Online]. Available: http://arxiv.org/abs/1904.07734

[34] Y.-C. Hsu, Y.-C. Liu, A. Ramasamy, and Z. Kira, “Re-evaluating continual learning scenarios: A categorization and case for strong baselines,” 2018, arXiv:1810.12488. [Online]. Available: http://arxiv.org/abs/1810.12488

[35] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009. [Online]. Available: http://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[36] Y. Le and X. Yang, “Tiny ImageNet visual recognition challenge,” CS 231N, vol. 7, no. 7, p. 3, 2015.

[37] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, “Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition,” Neural Netw., vol. 32, pp. 323–332, Aug. 2012.

[38] L. Hu, X. Pan, H. Yan, P. Hu, and T. He, “Exploiting higher-order patterns for community detection in attributed graphs,” Integr. Comput.-Aided Eng., vol. 28, no. 2, pp. 207–218, 2021.

LINTING GUAN received the M.Sc. degree in computer science from the University of Kunming University of Science and Technology, China. He is currently pursuing the Ph.D. degree with the School of Electronics and Information Engineering, Tongji University, Shanghai, China. He is also working with the School of Information Engineering, Zhejiang Ocean University, Zhejiang, China. His main research interest includes incremental learning in neural networks, which constitutes the core topic of his Ph.D. research project.

YAN WU received the Ph.D. degree in traffic information engineering and control from Shanghai Tiedao University, China, in 1999. From 2000 to 2003, she worked as a Postdoctoral Research Fellow with the Department of Electric Engineering, Fudan University, China. She is currently a Full Professor and a Doctoral Advisor with the College of Electronics and Information Engineering, Tongji University, Shanghai, China. She has published more than 140 papers in important national and international journals and conference proceedings. She is currently engaged in deep learning, image processing, pattern recognition, and self-driving.