ILCOC: An Incremental Learning Framework based on Contrastive One-class Classifiers

Wenju Sun Jing Zhang Danyu Wang Yangli-ao Geng Qingyong Li*
Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University, Beijing, 100044, China
{SunWenJu,j_zhang,WangDanYu,gengyla,liqy}@bjtu.edu.cn

Abstract

*In the class incremental learning, the number of classes to be handled dynamically raises with the number of considered tasks. The main challenge of this learning schema is catastrophic forgetting, that is the performance degradation on old tasks after learning new tasks. Existing incremental learning algorithms generally choose to train a multi-class classifier (e.g. softmax classifier), which learns a decision boundary to divide the feature space into several parts. Therefore, when new data arrive, the learned boundary will be updated and thus may cause forgetting. Compared with multi-class classifiers, a one-class classifier focuses on characterizing the distribution of a single class. As a result, the decision boundary learned for each category is tighter and does not change during learning new tasks. Inspired by this characteristic of one-class classifier, we propose a novel **I**ncremental **L**earning framework based on **C**ontrastive **O**ne-class **C**lassifiers (ILCOC) to avoid catastrophic forgetting. Specifically, we train a specific one-class classifier for each category and parallelly use them to achieve incremental multi-class recognition. Besides, we design a scale-boundary loss, a classifier-contrastive loss and a negative-suppression loss to strengthen the comparability of classifiers outputs and the discrimination ability of each one-class classifier. We evaluate ILCOC on MNIST, CIFAR-10 and Tiny-ImageNet datasets, and the experimental results show that ILCOC achieves state-of-the-art performance.*[1]

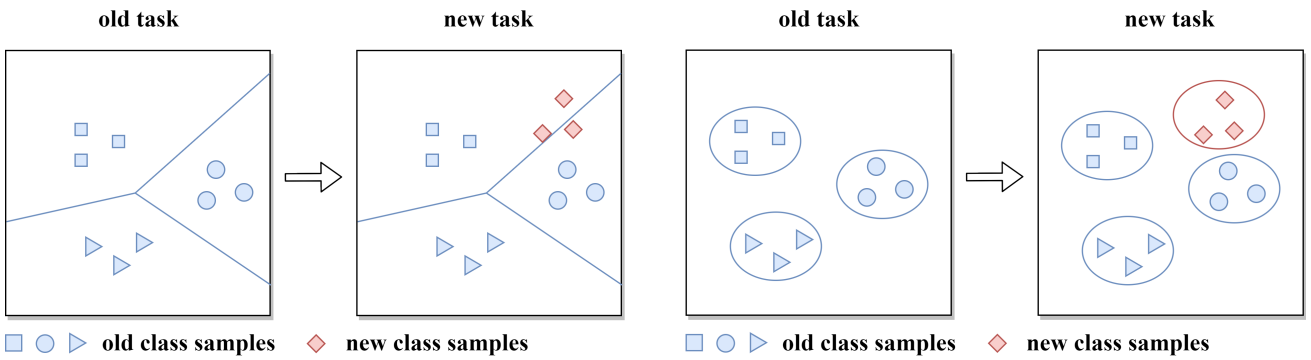
1. Introduction

Deep learning methods have been widely applied to many fields and have shown powerful performance in many tasks, such as image classification [14], object detection [26] and semantic segmentation [27]. Although it is great progress, current deep learning algorithms are still far from real intelligence. One of the main reasons is that most deep learning algorithms can not continuously learn new tasks while keeping the knowledge learned from old tasks. When a new task arrives, normal deep learning models need to be retrained with both new and old data. This training schema limits the application of deep learning techniques as, in some scenarios, the data of all tasks is not available at the same time. To solve this problem, incremental learning was proposed and has become a research hotspot [5].

Incremental learning tries to solve the problem of learning from a non-i.i.d. stream of data, with the goal of preserving and extending the acquired knowledge [19]. The main challenge of incremental learning is catastrophic forgetting [22, 7], which is manifested as that the performance on previous tasks drops dramatically due to the learning of new tasks. Particularly, there are three settings in the current incremental learning community: class incremental learning (class-IL), task incremental learning and domain incremental learning [35]. Our method and the subsequent discussion in this paper are all based on the class-IL setting. Under the class-IL setting, samples are divided into several sets with disjoint label spaces, and one-by-one use them to train the model in the task order.

To achieve incremental learning, various methods have been proposed. According to the working mechanism, existing incremental learning algorithms can be divided into three categories: rehearsal-based methods, regularization-based methods and dynamic architecture methods. Rehearsal-based methods need to use an additional memory space to store representative samples of previous tasks [25, 11, 8, 36]. Regularization-based algorithms use knowledge distillation technique to regularize network activation [16, 6] or penalize the changes of essential network parameters to avoid forgetting [39, 1, 12]. Dynamic architecture algorithms address the catastrophic forgetting problem by changing the network structure for specific tasks. Representatively, PackNet [21] and Piggyback [20] assign network parameters to each task by learning masks during training. PNN [30], DEN [38] and RCL [37] focus on expanding network capacity for new tasks.

*Corresponding author: Qingyong Li (liqy@bjtu.edu.cn).
[1]Code available at https://github.com/SunWenJu123/ILCOC.



(a) Multi-class Classifier

(b) One-class Classifier

Figure 1: The comparison between multi-class classifiers (a) and one-class classifiers (b). (a) The decision boundary of the multi-class classifier learned by old samples cannot be applied to new samples. (b) The decision boundaries of the one-class classifiers learned by old samples can be fused with the boundaries learned by new samples naturally.

A common point of the above methods is that they generally use multi-class classifier (e.g. softmax classifier) to achieve incremental classification, which has the following two limitations: (1) As old data is unavailable in new tasks, the imbalance between old and new data will lead incremental learning models to obtain a local optimal solution rather than a global one [23, 39]. As a result, the trained multi-class classifier based models are likely to recognize old samples as new categories [36, 40]. (2) As shown in Figure 1a, the working mechanism of multi-class classifiers is to learn decision boundaries to divide the entire embedding space into several parts. When new samples (red samples) arrive, the boundaries decided by old samples (blue samples) are likely to be unsuitable for the distribution of the new data. Compared with multi-class classifiers, one-class classifiers focus on describing the distribution of their target samples, which can be seen as aiming to find a minimum hypersphere to wrap all target samples. The decision boundaries of old classes learned in this way can be naturally merged with those of new classes, as shown in Figure 1b, which means the knowledge learned in old tasks will not be forgotten when learning new tasks. Inspired by this idea, we propose to use one-class classification models as the basic classifier to solve the incremental learning problem.

In this paper, we propose an **I**ncremental **L**earning framework based on **C**ontrastive **O**ne-class **C**lassifiers (ILCOC). ILCOC train multiple one-class classifiers with the same network structure but different parameters for all categories. Each one-class classifier is trained contrastively with the previous one-class classifiers. During testing, each one-class classifier calculates a confidence score which represents the probability that the input sample belongs to the corresponding category.

In this paper, our main contributions are listed as follows:

- We propose a novel incremental learning framework based on contrastive one-class classifiers named ILCOC, which avoid the catastrophic forgetting problem by parallelly using one-class classifiers.
- For incremental learning scenario, we design a scale-boundary loss, a classifier-contrastive loss and a negative-suppression loss to strengthen the comparability of classifiers and improve the discrimination ability of each one-class classifier, respectively.
- We evaluate ILCOC on MNIST, CIFAR-10 and Tiny-ImageNet datasets, and the experimental results show that our method achieves state-of-the-art performance.

The rest of this paper is organized as follows. In Section 2, we introduce some classic and latest algorithms of incremental learning and one-class learning. In Section 3, we propose ILCOC which prevents forgetting via one-class classifiers. In Section 4, we compare ILCOC with state-of-the-art methods and analyze the effectiveness of each part of ILCOC through ablation experiments. Finally, in Section 5, we summarize our work and offer some directions for future research.

2. Related work

2.1. Incremental Learning

According to the working mechanism, incremental learning algorithms can be divided into three categories: rehearsal-based methods, regularization-based methods and dynamic architecture methods.

Rehearsal-based methods utilize a fix sized memory to store samples of previous tasks and jointly train a neural network by using both stored and new data. iCaRL [25] com-

pletes multi-classification tasks by the prototypes, and innovatively utilizes samples in memory to generate the prototypes after learning each task. PRS [11] proposes a memory management algorithm to conquer the problem that data imbalance in memory, which is caused by training data that follow a long-tail distribution. Remind [8] uses compression algorithms to store more samples in the memory with a fixed size. BiC [36] uses the samples in the memory to calibrate the network output, which solves the problem of bias in the neural network. Although an addition memory effectively alleviates the forgetting caused by sample imbalance, this kind of algorithm limits its application due to its dependence on memory, especially in scenarios with small memory (mobile phones, personal computers).

Regularization-based methods focus on using regularization terms in the loss function to prevent forgetting. WA [40] solves the bias problem of the network by normalizing the parameters of the last FC layer. EWC [12], SI [39] and MAS [1] employ the fisher matrix, the contribution of loss reduction and the first-order derivative to identify important parameters, respectively. Then, all these three methods impose penalties on the changes of essential parameters to prevent forgetting. GEM [18] prevents forgetting by constraining the descent direction on the new task and the gradient direction on the old task to be an acute angle. Knowledge distillation based methods such as LWF [16] and PodNet [6] regularize the activation of the network to prevent forgetting.

Dynamic architecture methods provide independent parameters for each task to prevent forgetting. PNN [30] creates an independent network for each task, and transfers knowledge among different networks through horizontal connections. To decrease the number of network parameters, DEN [38] and RCL [37] dynamically expand the network capacity as needed when learning new tasks. PackNet [21] and CPG [10] use network compression technique to retain important parameters while improving the utilization rate of the network parameters. Piggyback [20] learns masks for each task to complete multiple tasks through a single network. Most of these algorithms have the advantage of dynamically expanding network capacity. Under class-IL setting, a model can not access the task identities for each sample during testing. However, due to the need of selecting network parameters for a corresponding task, some algorithms of this category may need to use task identifications when inferring, which violates the class-IL setting.

Our ILCOC method trains an independent one-class classifier for each class, so it belongs to the third category. Besides, due to the outputs among different tasks can be integrated naturally, ILCOC can meet the class-IL setting.

2.2. One-class learning

The goal of one-class learning is to learn a representation and/or a classifier that enables the recognition of positively labeled queries during inference, by using data with positive class and some quantity of weakly distributed negative class [24, 3], similar tasks include anomaly detection, outlier detection. In traditional methods, one-class SVM [31] and SVDD [34] learn a hypersphere with the smallest volume in the feature space to wrap the positive samples, and judge a sample is normal or not by whether the sample is within the hypersphere. Isolated forest [17] continuously separates samples by using a binary tree, and the sample that is easily separated is regarded as an anomaly. Deep-SVDD [28] and its semi-supervised version Deep-SAD [29] map a sample from the feature space to a latent space through a deep neural network and obtain a latent vector. This algorithm first initializes a fixed center point in the latent space, and then constrains the latent vector of a positive sample to be close to the center point during training. By assuming that the latent vectors of abnormal samples are far from the center, the distance between a latent vector and the center point can be regarded as an anomaly score.

3. Method

In this section, we will introduce the proposed ILCOC method in detail. Specifically, we first introduce the incremental learning setting and notations of this paper in Section 3.1. After that, a basic incremental learning framework based on parallel one-class classifiers is presented in Section 3.2. Based on this framework, we propose our ILCOC method in Section 3.3 which has a more powerful performance.

3.1. Problem Definition

This paper focuses on the class-IL setting, in which the entire dataset D is divided into disjoint T parts, that is $D = \{D^t\}_{t=1}^{T}$. Under this setting, the data of task t can be denoted as $D^t = \{(\mathbf{x}_k^t, y_k^t)\}_{k=1}^{n_t}$, where n_t, $\mathbf{x}_k^t$ and y_k^t represent the number of samples in D^t, the feature and the label of the k-th sample, respectively. Let Y^t represent the label set of D^t, and it is worth noting that there is no intersection between the label sets of different tasks, that is, when $i \neq j$, $Y^i \cap Y^j = \varnothing$. Taking the MNIST [15] as an example, we can divide it into 5 subsets, and each subset contains two categories of samples. Accordingly, their label sets can be denoted as $Y^1 = \{0, 1\}$, $Y^2 = \{2, 3\}$, $Y^3 = \{4, 5\}$, $Y^4 = \{6, 7\}$, $Y^5 = \{8, 9\}$.

Each data subset is divided into training and testing set with a same manner $D^t = \{D_{train}^t, D_{test}^t\}$. During the learning of task t, the model learns from the training set D_{train}^t. Then, we evaluate the model by using $D_{test}^1 \cup D_{test}^2 ... \cup D_{test}^t$, which means the performance of the

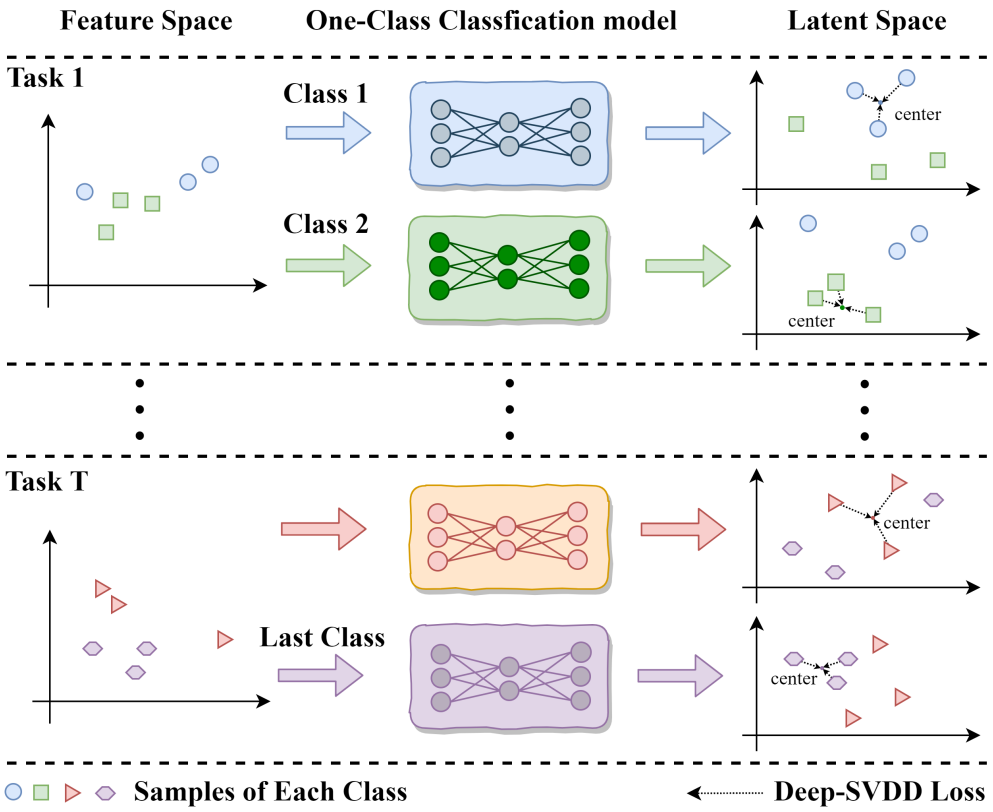


Figure 2: The diagram of incremental learning framework based on parallel one-class classifiers.

model on all learned tasks is tested. Note that our algorithm does not utilize the data of previous tasks when training a new task. Besides, we follow the class-IL setting where the model has no access to the task identification during testing.

3.2. Incremental Learning Based on Parallel One-Class Classifiers

The incremental learning models based on multi-class classifiers often face the following two challenges: (1) Due to the imbalance between old and new data, network parameters will be updated to fit the distribution of new categories. Under this situation, the predictions of multi-class classifiers are biased to new classes [36, 40], which may cause catastrophic forgetting. (2) Multi-class classifiers learn decision boundaries to completely divide the feature space, which may be not suitable for new tasks, as shown in Figure 1a. Thus, training on new tasks inevitably changes the decision boundaries learned by old tasks.

Compared with multi-class classifiers, one-class classifiers are robust to the scenarios with imbalanced samples. Besides, as shown in Figure 1b, one-class classifiers focus on finding a minimal hypersphere to wrap all target samples, which means their decision boundaries are tighter and do not affect the modeling of other samples. Thus, we propose to use one-class classifiers to achieve incremental recognition.

Figure 2 illustrates the diagram of our incremental learning framework based on parallel one-class classifiers. In this framework, we train a specific one-class classification model for each category, and then use their outputs to represent the probabilities that samples belong to their corresponding classes. As it is a parallel structure, we call it **I**ncremental **L**earning based on **P**arallel **O**ne-class **C**lassifiers (ILPOC) framework. Considering the performance and scalability, we choose Deep SVDD [28] as our basic one-class classification model to implement our multi-class classification idea. The core principle of Deep SVDD is to use a neural network to project samples from a visual feature space to a latent space, in which the samples belonging to the target category are required to be as close as possible to a center point. Specifically, for the new category i of task t, the objective of Deep SVDD model is:

$$\min_{W_i} \frac{1}{n_t^+} \sum_{y_k^t=i} \left\| \phi(\mathbf{x}_k^t; W_i) - \mathbf{c}_i \right\|^2 + \lambda \left\| W_i \right\|_F^2, \quad (1)$$

where $\phi(.)$, W_i and $\mathbf{c}_i$ represent the projection function, its parameters and the center point for category i, respectively.

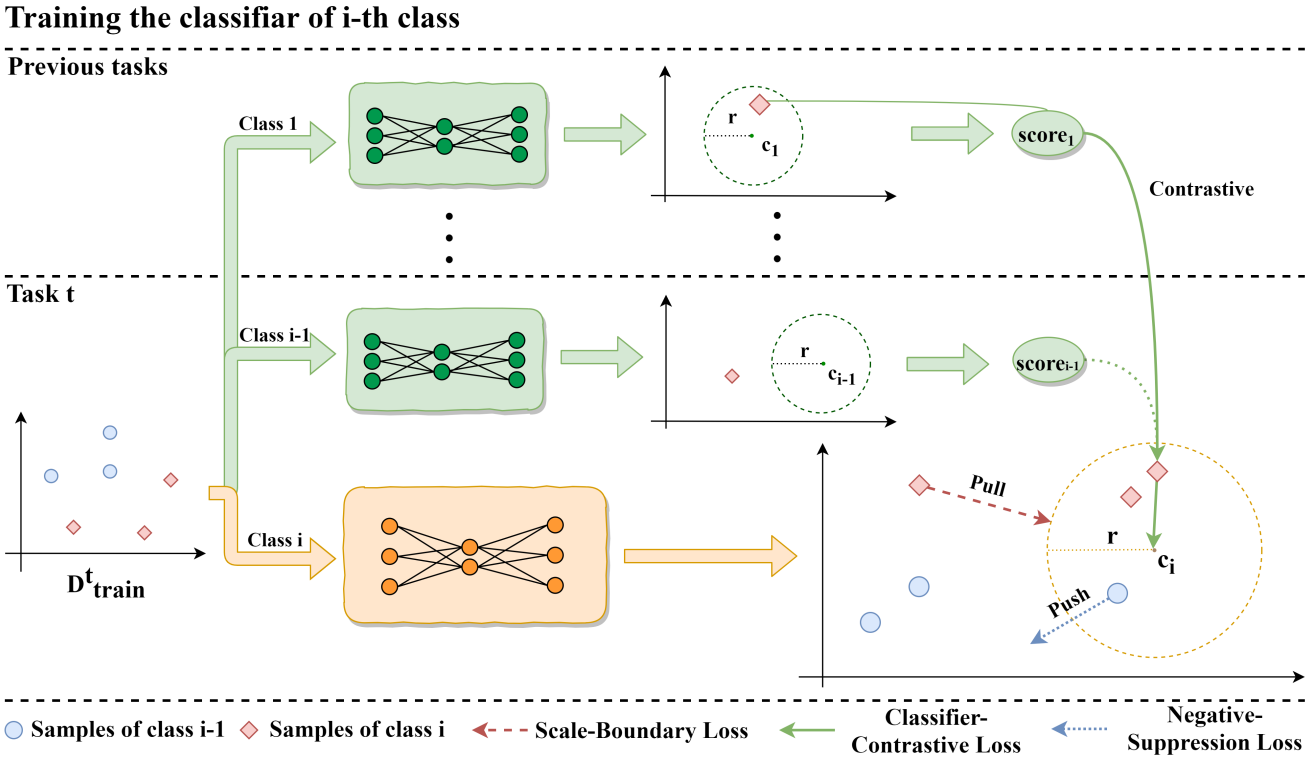


Figure 3: The diagram of ILCOC. During the training of i-th category in task t, the scale-boundary loss (red arrow) pulls all samples of class i into the hypersphere. All the hyperspheres in each latent space have the same radius r, which ensures the outputs of all models following similar distribution. Besides, the classifier-contrastive loss (green arrow) forces the model to output a score that higher than the scores predicted by the previous model. On the other hand, for the samples that belong to other classes, the negative-suppression loss (blue arrow) pushes the latent embedding vectors far away from the center.

$n_t^+ = |\{\mathbf{x}_k^t | (\mathbf{x}_k^t, y_k^t) \in D_{train}^t \ and \ y_k^t = i\}|$ is the number of samples belonging to category i in set D_{train}^t. The first term requires the latent vectors of samples to be as close as possible to the center $\mathbf{c}_i$. The second term is a regularization term with a weight parameter $\lambda > 0$. In our framework, when a new task t comes, we will train $|Y^t|$ Deep SVDD models.

After training, we calculate the score of the sample $\mathbf{x}$ by the following formula:

$$score_i(\mathbf{x}) = \frac{1}{\|\phi(\mathbf{x}; W_i) - \mathbf{c}_i\|^2 + \varepsilon}, \tag{2}$$

where $score_i(\mathbf{x})$ can be seen as the confidence that the sample $\mathbf{x}$ belongs to the category i, and ε is set to prevent the denominator equal to 0. Directly, we choose the category with the highest score as the prediction result of $\mathbf{x}$, that is:

$$\hat{y} = \arg\max_i \{score_i(\mathbf{x})\}_{i=1}^{|Y|}, \tag{3}$$

where $\hat{y}$ represents the prediction of $\mathbf{x}$, and $Y = Y^1 \cup Y^2 \cup \ldots \cup Y^t$ represents the set of all seen categories.

3.3. Incremental Learning Based on Contrastive One-Class Classifiers

The ILPOC framework in Section 3.2 has two major limitations: (1) The comparability of the output scores from different one-class classification models is inadequate. Due to the difference in sample distribution and network parameters, the distribution of output scores is quite different. (2) As each one-class classification model can only see the data in a single task, the trained model may have poor performance on unseen data.

To mitigate the impact of the above two problems, we further design a **scale-boundary loss**, a **classifier-contrastive loss** and a **negative-suppression loss**. Figure 3 shows the diagram of our method with the above three loss functions.

Scale-boundary loss. In order to improve the comparability of sample scores from different models, we require that the target samples of different classification models have a similar distribution in each latent embedding space.

Specifically, we define a hypersphere whose center is $\mathbf{c}_i$ and radius is r, and ask the positive samples should be in this hypersphere. This practice can not only relax the constraint in Eq. (1) to avoid the model over-fitting, but also make positive samples have a similar distribution pattern. The formula of the scale-boundary loss is:

$$L_{SB} = \frac{1}{n_t^+} \sum_{y_k^t=i} max\{0, \left\|\phi(\mathbf{x}_k^t; W_i) - \mathbf{c}_i\right\|^2 - r^2\}, \quad (4)$$

where r is a fixed hyperparameter shared by all one-class classification models.

Classifier-contrastive loss. As each one-class classification model can only see the data of one task, the trained model likely has poor discrimination performance on the data of other tasks, especially those who are similar to the samples in this task. To solve this problem, we propose a classifier-contrastive loss to strengthen the discrimination ability of newer models. Given the samples of category i in the task t, the classifier-contrastive loss for the i-th model is:

$$\begin{aligned} L_{CC} = & \frac{1}{n_t^+} \sum_{y_k^t=i} (\frac{1}{|Y_i^p|} \sum_{j \in Y_i^p} max\{0, \left\|\phi(\mathbf{x}_k^t; W_i) - \mathbf{c}_i\right\|^2 \\ & - \left\|\phi(\mathbf{x}_k^t; W_j) - \mathbf{c}_j\right\|^2\}), \end{aligned} \quad (5)$$

where $Y_i^p = \{j | 0 \leq j < i\}$ contains the indexes of the models trained in the former tasks. The classifier-contrastive loss forces the new one-class classification model to give higher confidence scores for the positive samples than those calculated by the models trained before.

Negative-suppression loss. The scale-boundary loss and the classifier-contrastive loss are applied to the samples belonging to the current category i. But we need to notice that there are also other categories in each task, which contain abundant information that can be used to improve the discrimination ability of a one-class classification model. To make full use of these negative (non-target) data, we propose a simple negative-suppression loss to suppress their scores, which can be denoted as:

$$L_{NS} = \frac{1}{n_t^-} \sum_{y_k^t \neq i} \frac{1}{\left\|\phi(\mathbf{x}_k^t; W_i) - \mathbf{c}_i\right\|^2 + \varepsilon}, \quad (6)$$

where $n_t^- = |\{\mathbf{x}_k^t | (\mathbf{x}_k^t, y_k^t) \in D_{train}^t \ and \ y_k^t \neq i\}|$ is the number of samples not belonging to category i in set D_{train}^t.

Finally, the total loss of training a one-class classification model is as follows:

$$L = L_{SB} + \alpha_1 L_{CC} + \alpha_2 L_{NS}, \quad (7)$$

where α_1 and α_2 are weight parameters.

4. Experiments

We compare our ILCOC method with state-of-the-art methods on MNIST, CIFAR-10 and Tiny-ImageNet datasets. We also perform ablation experiments to analyze the effect of each component in our approach.

4.1. Datasets and Implementation Details

We carry out experiments following the experimental settings in [4]. We use the widely used precision as our evaluation metric and three datasets are selected as our benchmarks:

MNIST [15] includes 60,000 training images and 10,000 testing images of 10 handwritten digit classes. These 10 classes are divided into 5 incremental batches.

CIFAR-10 [13] contains 60,000 32×32 RGB images from 10 different categories. The 10 classes are also divided into 5 incremental batches.

Tiny-ImageNet [33] has 200 classes, and each class has 500 training images, 50 validation images and 50 testing images. The 200 classes are divided into 10 incremental batches.

All methods are implemented with PyTorch, and optimized by stochastic gradient descent. For MNIST dataset, we employ a fully-connected network to achieve feature projection. The network has two hidden layers and each layer contains 100 neurons. For CIFAR-10 and Tiny-ImageNet datasets, we use ResNet18 [9] as our basic network, and train it with 50 and 100 epochs respectively.

For the hyperparameters in ILCOC, we set $r^2 = 0.1$ in all three datasets. We set $\alpha_1 = 1$ and $\alpha_2 = 0.8$ for MNIST and CIFAR-10 datasets. For Tiny-ImageNet, we set $\alpha_1 = 0.5$ and $\alpha_2 = 0.04$.

4.2. Incremental Learning Comparison

In this section, we compare the performance of ILCOC with state-of-the-art methods on three datasets with different scales (MNIST, CIFAR-10 and Tiny-ImageNet). We select three regularization-based methods which are oEWC [32], SI [39] and LWF [16]. oEWC [32] and SI [39] use the fisher matrix and the contribution of loss reduction to measure the importance of network parameters and penalize their changes, respectively. LWF [16] regularizes activation by knowledge distillation to prevent forgetting. Besides, we choose a rehearsal-based algorithm FDR [2] as a contrast. It defines a function space with the help of samples in the memory, and uses the function space to measure the importance of network parameters. After then, it will penalize the changes of the essential parameters during learning new tasks. Note that, compared with other methods, FDR needs an additional memory which can store up to 200 samples. For the convenience of comparison, we also provide two basic comparison methods which are JOINT and SGD. JOINT

Method	MNSIT	CIFAR-10	Tiny-ImageNet
JOINT	95.57%	92.20%	95.99%
SGD	19.60%	19.62%	7.92%
oEWC [32]	20.46%	19.49%	7.58%
SI [39]	19.27%	19.48%	6.58%
LWF [16]	19.62%	19.61%	8.46%
FDR* [2]	**79.43%**	30.91%	8.70%
Ours(ILCOC)	74.51%	**38.40%**	**16.97%**

Table 1: Incremental learning performance comparison on three datasets. The results of the comparison methods come from [4]. Note that FDR* needs additional memory to store old samples.

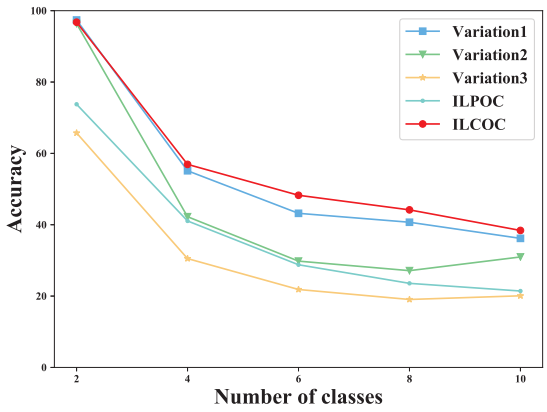


Figure 4: The results of ablation experiments on CIFAR-10 dataset. The x-axis indicates the number of learned classes.

uses all task data to fully train a neural network, which can be seen as the upper bound of incremental learning; SGD does not use any incremental learning strategy, and the network is directly fine-tuned with new data in each task.

Table 1 reports the test accuracies of all methods on three incremental learning benchmarks. The results show that our ILCOC achieves state-of-the-art performance in almost all datasets. Specifically, compared with oEWC, SI and LWF, the accuracy of ILCOC is much higher, verifying our idea that one-class classifiers can produce robust decision boundaries and help the model alleviate the impact of catastrophic forgetting. Besides, we can also observe that ILCOC has better performance than FDR, which uses an additional memory to store representative samples. We think the reason may be that the fixed memory size of FDR can not ensure the stored samples simulate the data distribution of true samples.

4.3. Ablation Study

Based on the architecture of ILPOC, ILCOC adds a scale-boundary loss, a classifier-contrastive loss and a negative-suppression loss to establish the relationship between one-class classifiers and enhance its feature learning ability. In this subsection, we analyze the effect of each component of ILCOC by ablation experiments. The variations we construct are as follows:

Variation1: we set $r = 0$ on the basis of ILCOC, which means that the scale-boundary loss is degraded to Eq. (1);

Variation2: we set $\alpha_1 = 0$ on the basis of ILCOC, that is, the classifier-contrastive loss has no effect;

Variation3: we set $\alpha_2 = 0$ on the basis of ILCOC, which means the negative-suppression loss is removed;

ILPOC: the basic ILPOC model which is obtained by setting $r = 0$, $\alpha_1 = 0$, $\alpha_2 = 0$ on the basis of ILCOC.

The results of ablation experiments are shown in Figure 4. First of all, comparing Variation1 and ILCOC, it can be seen that Variation1 obtains a higher accuracy in task 1 (when the number of learned classes is two). This is because the scale-boundary loss slightly suppresses the performance of one-class classifiers. We also can find that the accuracies of ILPOC in subsequent tasks are lower than ILCOC, which shows that the scale-boundary loss can improve the comparability of different one-class classifiers. Secondly, the performance gap between Variation2 and ILCOC proves that the classifier-contrastive loss can effectively use the information of former classifiers to help the training of the current classifier. Thirdly, the results of Variation3 and ILCOC show that the negative-suppression loss successfully improves the discrimination ability of each one-class classifier. Finally, by comparing ILPOC and ILCOC, we can infer that the added three loss functions greatly enhance the multi-classification performance of the ILPOC model. All these results prove the effectiveness and robustness of our ILCOC method.

5. Conclusion

In this paper, we introduce an incremental learning framework based on contrastive one-class classifiers which we called ILCOC. ILCOC parallelly uses one-class classifiers to produce tight decision boundaries, which can be naturally merged together. To improve the comparability of classifier outputs and enhance the relationship between one-class classifiers, we further design a scale-boundary loss, a classifier-contrastive loss and a negative-suppression loss. The experimental results show that ILCOC can achieve state-of-the-art performance in most cases. Since ILCOC is a parallel structure, the size of the model increases linearly with the number of tasks. In the future, we will enhance the capacity of ILCOC in this aspect.

Acknowledgment. This work was supported in part by

the National Natural Science Foundation of China under Grant U2034211, 62006017, in part by the Fundamental Research Funds for the Central Universities under Grant 2020JBZD010, in part by the Beijing Natural Science Foundation under Grant L191016.

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Eur. Conf. Comput. Vis.*, pages 139–154, 2018. 1, 3

[2] Ari S. Benjamin, David Rolnick, and Konrad P. Körding. Measuring and regularizing networks in function space. In *International Conference on Learning Representations*, 2019. 6, 7

[3] Jay Bhatt and Nikita S Patel. A survey on one class classification using ensembles method. *IJIRST-International Journal for Innovative Research in Science and Technology*, 1:19–23, 2014. 3

[4] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and SIMONE CALDERARA. Dark experience for general continual learning: a strong, simple baseline. In *Adv. Neural Inform. Process. Syst.*, volume 33, pages 15920–15930, 2020. 6, 7

[5] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1

[6] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Eur. Conf. Comput. Vis.*, pages 86–102, 2020. 1, 3

[7] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013. 1

[8] Tyler L Hayes, Kushal Kafle, Robik Shrestha, Manoj Acharya, and Christopher Kanan. Remind your neural network to prevent catastrophic forgetting. In *Eur. Conf. Comput. Vis.*, pages 466–483, 2020. 1, 3

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016. 6

[10] Ching-Yi Hung, Cheng-Hao Tu, Cheng-En Wu, Chien-Hung Chen, Yi-Ming Chan, and Chu-Song Chen. Compacting, picking and growing for unforgetting continual learning. In *Adv. Neural Inform. Process. Syst.*, volume 32, pages 13647–13657, 2019. 3

[11] Chris Dongjoo Kim, Jinseo Jeong, and Gunhee Kim. Imbalanced continual learning with partitioning reservoir sampling. In *Eur. Conf. Comput. Vis.*, pages 411–428, 2020. 1, 3

[12] 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. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 1, 3

[13] Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012. 6

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inform. Process. Syst.*, 25:1097–1105, 2012. 1

[15] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. 3, 6

[16] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(12):2935–2947, 2017. 1, 3, 6, 7

[17] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *IEEE International Conference on Data Mining*, pages 413–422. IEEE, 2008. 3

[18] David Lopez-Paz and Marc' Aurelio Ranzato. Gradient episodic memory for continual learning. In *Adv. Neural Inform. Process. Syst.*, volume 30, pages 6467–6476, 2017. 3

[19] Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *arXiv preprint arXiv:2101.10423*, 2021. 1

[20] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Eur. Conf. Comput. Vis.*, pages 67–82, 2018. 1, 3

[21] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7765–7773, 2018. 1, 3

[22] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989. 1

[23] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh. Understanding the role of training regimes in continual learning. In *Adv. Neural Inform. Process. Syst.*, volume 33, pages 7308–7320, 2020. 2

[24] Pramuditha Perera, Poojan Oza, and Vishal M Patel. One-class classification: A survey. *arXiv preprint arXiv:2101.03064*, 2021. 3

[25] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2001–2010, 2017. 1, 2

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Adv. Neural Inform. Process. Syst.*, volume 28, pages 91–99, 2015. 1

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241, 2015. 1

[28] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International Conference on Machine Learning*, pages 4393–4402, 2018. 3, 4

[29] Lukas Ruff, Robert A. Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. Deep semi-supervised anomaly detection. In *International Conference on Learning Representations*, 2020. 3

[30] 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. 1, 3

[31] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001. 3

[32] 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*, pages 4528–4537, 2018. 6, 7

[33] Stanford. Tiny imagenet challenge (cs231n). 2015. 6

[34] David MJ Tax and Robert PW Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004. 3

[35] Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019. 1

[36] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 374–382, 2019. 1, 2, 3, 4

[37] Ju Xu and Zhanxing Zhu. Reinforced continual learning. In *Adv. Neural Inform. Process. Syst.*, volume 31, pages 907–916, 2018. 1, 3

[38] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations*, 2018. 1, 3

[39] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pages 3987–3995, 2017. 1, 2, 3, 6, 7

[40] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 13208–13217, 2020. 2, 3, 4