Multi-layer Rehearsal Feature Augmentation for Class-Incremental Learning

Bowen Zheng [1 2] **Da-Wei Zhou** [1 2] **Han-Jia Ye** [1 2] **De-Chuan Zhan** [1 2]

Abstract

Class-Incremental Learning (CIL) seeks to learn new concepts without forgetting previously learned knowledge. To achieve this, rehearsal-based methods keep a replay memory consisting of a small number of trained samples from previous tasks. However, recent studies show that rehearsal-based methods are prone to overfitting on rehearsal samples, resulting in poor generalization on previous tasks. Since the generalization error is bounded by the margin on the training dataset, in this paper, we study the generalization by all-layer margin on deep neural networks to alleviate catastrophic forgetting. Specifically, we show that the average margin of the rehearsal samples are smaller during incremental learning. To acquire larger margin thus better generalization on rehearsal samples, we propose Multi-layer Rehearsal Feature Augmentation (MRFA) in rehearsal training to optimize the all-layer margin on rehearsal samples. The proposed method augments the features of rehearsal samples at each layer by gradient ascent step of the current model with respect to the feature. With such augmentations on layer features, the margin on rehearsal samples are larger, rehearsal samples are able to provide more information for refining the decision boundary during incremental learning, thus alleviating catastrophic forgetting. Extensive experiments show the effectiveness of MRFA on various CIL scenarios.

1. Introduction

In traditional supervised learning, the training samples are assumed to be drawn i.i.d. from a stationary distribution. However, in real-world situations, this assumption does not hold since new concepts and knowledge increase over time. It is necessary for learning systems to adapt to new knowledge while keeping the previously learned knowledge. This motivates the research in Continual Learning (or Incremental Learning). *Class-Incremental Learning* (CIL) (Zhou et al., 2023b) is one of the scenarios where new concepts incrementally emerge as new classes. The main challenge for Continual Learning is *catastrophic forgetting* (McCloskey & Cohen, 1989), where the learning system usually forgets previously learned knowledge fast and catastrophically. The forgetting is caused by the distribution shift on the input samples and labels, especially at the task boundary where the rapid shift occurs. When the distribution shift occurs, the gradients, intermediate representations are biased towards the new task, causing the test accuracy drop on old tasks.

One of the widely adopted strategies to avoid catastrophic forgetting is to keep a small number of samples from previously learned tasks for replay. Such methods are called rehearsal-based approaches (Robins, 1995; Rebuffi et al., 2017; Chaudhry et al., 2019; Liu et al., 2020). This strategy is generally effective since it mitigates the distribution shift during new task training. However, recent studies show that rehearsal-based methods are prone to overfitting on rehearsal samples (Verwimp et al., 2021). When the model is training with additional rehearsal samples, the gradients are initially dominated by the new task because of the significant loss gap between rehearsal samples and training samples from the new task. After some epochs, their losses are balanced, the model will end up at a high-loss ridge, harming the generalization on previous tasks.

There are various ways to bound the generalization error, margin-based generalization bounds are important for classification models (Koltchinskii & Panchenko, 2002; Kakade et al., 2008). All-layer margin (Wei & Ma, 2020) is one of the margin-based generalization bounds for deep neural networks. It is defined as the minimal perturbation required to alter the prediction of the model for a sample. Generally, large margin means good generalization. It is also true for all-layer margin, the larger the minimal perturbation is, the larger the margin is, thus better generalization the model will achieve. All-layer margin offers a finegrained and tractable way to estimate the generalization error for deep networks. Therefore, we wonder if we can reveal the loss of generaliza-

[1]School of Artificial Intelligence, Nanjing University, China [2]National Key Laboratory for Novel Software Technology, Nanjing University, China. Correspondence to: Da-Wei Zhou <zhoudw@lamda.nju.edu.cn>, Han-Jia Ye <yehj@lamda.nju.edu.cn>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

tion on rehearsal samples by studying the evolution of the all-layer margin on rehearsal samples, then try to alleviate catastrophic forgetting by enlarging the margin.

In this paper, we reveal the overfitting on rehearsal samples from another perspective by studying how the all-layer margin of the network evolves in CIL. Specifically, we investigate the all-layer margin of the network quantitatively with its upper bound and show the decision boundary visually. As a result, we find that the margin shrinkage indeed happens in rehearsal samples. Therefore, we can alleviate the overfitting on the rehearsal samples by making the margin larger. Furthermore, we propose multi-layer rehearsal feature augmentation (MRFA) for class-incremental learning. The proposed method augments the features of rehearsal samples at each layer by gradient ascent step of the current model with respect to the feature. With such augmentation on layer features, rehearsal samples are able to provide more information for refining the decision boundary during incremental learning, thus alleviating catastrophic forgetting. Moreover, extensive experiments are performed to show the effectiveness of MRFA on various settings of CIL.

Our contributions can be summarized as 1) We investigate the overfitting on rehearsal samples from the perspective of all-layer margin of the network, quantitatively and visually. 2) We propose MRFA to make the margin of the rehearsal samples larger, thus alleviating the forgetting of the network. 3) We perform extensive experiments to verify the effectiveness of the proposed method.

2. Related Works

2.1. Class-Incremental Learning

Class-Incremental Learning (CIL) is an incremental learning scenario where the model is learned task by task with a different set of classes. During inference, no task information about the samples is available. Many techniques and frameworks are proposed to alleviate catastrophic forgetting and improve the performance in CIL.

Rehearsal-based methods Buzzega et al. (2020) store exemplars of previous tasks and replay them in follow-up tasks. It makes the learned representation less forgetful by adjusting the input distribution towards the learned tasks. Many works focus on how to select exemplars (Rebuffi et al., 2017; Wu et al., 2019; Tiwari et al., 2022; Liu et al., 2020). Exemplars can also be obtained by generative models (Shin et al., 2017). However, recent studies show that rehearsal-based methods are prone to overfitting on rehearsal samples (Verwimp et al., 2021).

There are several works aiming to enhance the utilization of the rehearsal samples, which coincides with the root motivation of this paper. RAR (Kumari et al., 2022) pairs each rehearsal sample with the sample of the task, and interpolates them to generate marginal samples for rehearsal. GMED (Jin et al., 2021) use gradient-based methods to edit the raw image of the rehearsal samples, making the network focus more on the samples with the most increased loss.

There are works also consider augmentations on rehearsal samples (Zhang et al., 2022; Yang et al., 2023). Zhang et al. (2022) proposes to combine data augmentation with multiple iterations in online continual learning (OCL), and further adopts bootstrapped policy gradient strategy to automatically determine the number of iterations and the augmentation strength. although it studies the rehearsal samples, their methods do not apply to CIL, since it studies the balance between the number of iterations and the augmentation strength in OCL.

There are other works mentioned margin in CIL. Hou et al. (2019) proposes a margin ranking loss to separate the embeddings of old and new. However, the margin in this loss only considers the final output embedding of the deep neural network, which is different from the all-layer margin that considers all of the layers in this paper.

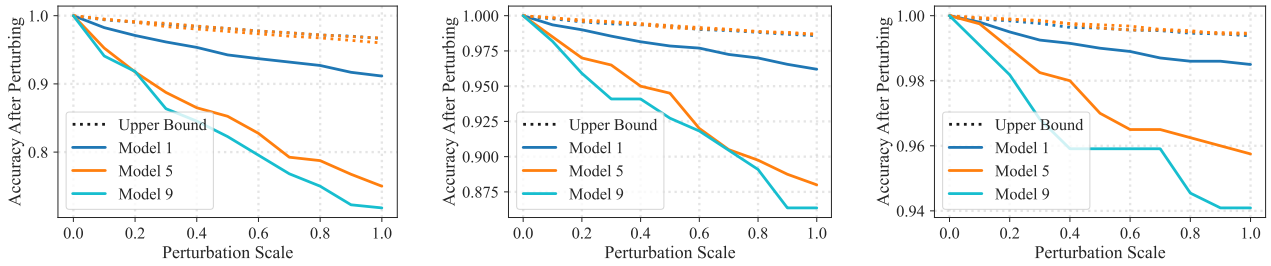
Overall, although the utilization of the rehearsal samples are studied in previous works, the margin on the rehearsal samples has never been explored and studies related to the margin are limited.

We discuss additional related works in Appendix A.

2.2. Margin and Generalization

Margin is an important notion throughout the machine learning history. Large margin principle produces remarkable and empirical results for classification (Vapnik, 1999) and regression (Drucker et al., 1996). For linear models, the generalization error can be easily bounded by normalized output margin (Koltchinskii & Panchenko, 2002; Kakade et al., 2008). However, for deep models, the generalization bounds are more complicated and requiring the normalization by a quantity that either scales exponentially in depth or depends on complex properties of the network (Neyshabur et al., 2015; Golowich et al., 2018). Therefore, generalization bounds on the input margin become the alternatives for deep networks. Sokolić et al. (2017) provides generalization bounds based on the input margin of the network, but the bounds depend exponentially on the dimension of the data manifold. Yan et al. (2019) optimizes the adversarial margin in the input space. All-layer margin (Wei & Ma, 2020) considers all of the layers simultaneously, avoiding explicit dependency on exponential exponential dependency on network depth.

To the best of our knowledge, the margin of the network on the rehearsal memory during CIL has never been explored. Our work aims to investigate the overfitting on rehearsal



(a) Block 1 (b) Block 2 (c) Block 3

Figure 1. Accuracy drop after different scales of perturbation. With the approximations in section 4.1, the all-layer margin is estimated by the rate of decline of the accuracy after input perturbations along a certain direction. *Model k* represents the model trained after task k, the model has trained on $k+1$ tasks $(0, \ldots, k)$. *Upper Bound* represents the accuracy drop on the samples from the current task. As we can see from the figures, samples in the rehearsal memory show more rapid accuracy decline with the same amount of perturbation scale as the task goes on. It happens for every layer of the network, which means **the margin of the rehearsal samples is averagely smaller as the model is incrementally updated for new tasks**.

samples from the perspective of the margin, providing better understanding about the catastrophic forgetting.

3. Preliminaries

In this section, we provide background knowledge about the problem formulation of class-incremental learning and the formal formulation of all-layer margin with its generalization bound for deep neural networks.

3.1. Problem Formulation

In CIL scenarios, we have multiple classification tasks to learn sequentially. Let $\mathcal{D}_t$ be the training dataset of the tth task. $(\boldsymbol{x}_i^{(t)}, y_i^{(t)}) \in \mathcal{D}_t$ is a sample. $\boldsymbol{x}_i^{(t)}$ is the input, $y_i^{(t)}$ is the label. Let $\mathcal{C}_t = \bigcup_i \{y_i^{(t)}\}$ be the class set of task t. In CIL, $\forall t_1 \neq t_2, \mathcal{C}_{t_1} \cap \mathcal{C}_{t_2} = \emptyset$. In each task, we only train the model on $\mathcal{D}_t$, but test on all the tasks the model has trained on, i.e., the seen tasks. For example, when the model is training on task t_i, the seen tasks are tasks $t_j (j \leq i)$. In rehearsal-based CIL, a small number of rehearsal samples $\mathcal{M}$ are allowed to be stored for later tasks. The total number of rehearsal samples should be constant during the CIL training. The goal is to make the model get better performance on all the seen tasks.

3.2. All-layer Margin

The margin is conventionally defined on the output space which equals the gap between predictions on the true label and the second confident label. For shallow network architectures, such margin produces great theoretical and empirical results for classification and regression. For example, kernel SVMs (Boser et al., 1992) has an analytical form for the output margin. However, for deep network architectures, the theoretical generalization bound is complicated and intractable to estimate (Golowich et al., 2018; Nagarajan & Kolter, 2019).

In order to overcome the limitations of the output margin on deep models, an alternative margin is defined on the input space (Elsayed et al., 2018) which measures how much perturbation a layer can resist to alter its final prediction for a sample. One of such margins with theoretical generalization bound is all-layer margin (Wei & Ma, 2020). Suppose the classification model $F(\boldsymbol{x}) = f_L \circ \cdots \circ f_1(\boldsymbol{x})$ is composed of L layers, and $\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_L$ are perturbations intended to be added to each layer's input. The margin is defined by the minimum norm of $\boldsymbol{\delta}$ required to alter the correct prediction:

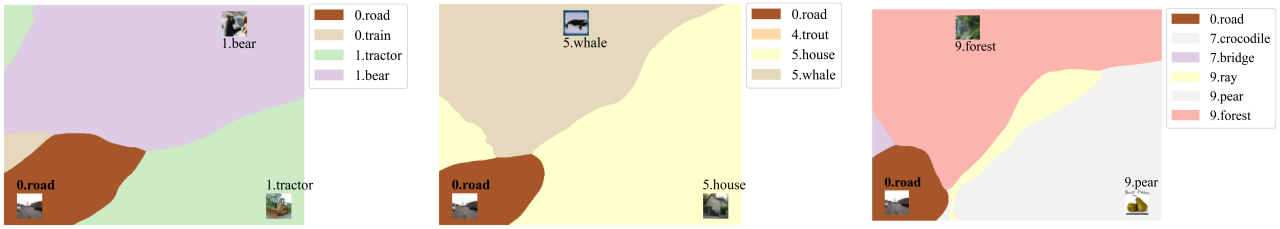
$$m_F(\boldsymbol{x}_i, y_i) := \min_{\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_L} \sqrt{\sum_{l=1}^{L} \|\boldsymbol{\delta}_l\|_2^2}, \quad \text{s.t. } \arg\max F(\boldsymbol{x}_i, \boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_L) \neq y_i, \tag{1}$$

where $(\boldsymbol{x}_i, y_i)$ is a pair of sample and label from the training set, $F(\boldsymbol{x}_i, \boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_L)$ is the model with perturbations $\boldsymbol{\delta}_l$ added at the input of layer l.

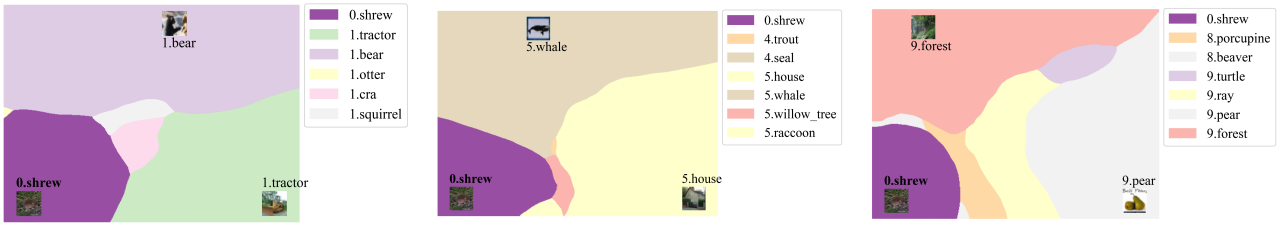
The generalization error can be bounded without explicit exponential dependency on network depth using all-layer margin (Wei & Ma, 2020). To state it formally, with probability $1-\delta$ over the draw of the training data, all of the classifiers $F \in \mathcal{F}$ that achieve training error 0 is proved to satisfy the following bound in terms of the all-layer margin:

$$\mathbb{E}_P \left[\ell_{0\text{-}1}(F(\boldsymbol{x}), y)\right] \lesssim \frac{\sum_i \mathcal{C}_i}{\sqrt{n}} \sqrt{\mathbb{E}_{(\boldsymbol{x},y)\sim P_n} \left[\frac{1}{m_F(\boldsymbol{x}, y)^2}\right]} \log^2 n + \zeta, \tag{2}$$

where $\zeta := O\left(\frac{\log(1/\delta) + \log n}{n}\right)$ is a low-order term. It offers the theoretical guarantee that with larger expected margin on the training set, we can achieve lower generalization error.



(a) Input Space of Model 1 (b) Input Space of Model 5 (c) Input Space of Model 9



(d) Input Space of Model 1 (e) Input Space of Model 5 (f) Input Space of Model 9

Figure 2. Evolution of decision boundary around rehearsal samples. The number leading the class name is the task number of the class. For example, *0.road* means class *road* is from task 0. The two sub-figure rows show the decision boundaries around two samples in rehearsal memory from the initial task, whose decision regions are in darker color (at the left bottom of each figure, i.e. *0.road* and *0.shrew* in two rows). The other two samples are from the current task, whose decision regions. Since the all-layer margin is formulated by the minimal amount of perturbation that alter the prediction of the network, the visualizations of decision boundary reflect the margin intuitively. **As the task goes on, the decision boundary shrinks towards the sample, which means it can no longer resist the same amount of perturbation, leading to smaller margin for rehearsal samples**.

4. Methodology

With the preliminaries above, the next issue we care about is that can we attribute the loss of generalization on old tasks to the smaller margin of the rehearsal samples? Does the margin shrinkage actually happen in rehearsal-based CIL? To find it out empirically, in this section, we first approximate the margin for better tractability. Then, with the approximations, we investigate how the all-layer margin of the network evolves during CIL training in two aspects. Finally, we propose our method.

4.1. Evolution of Margin in Rehearsal-based CIL

In this section, we approximate the margin for better tractability. The definition of all-layer margin is a min-max formulation, and contains multiple optimizable variables, which is hard to estimate. Alternatively, we turn to estimate its upper bound. In fact, all-layer margin is bounded by the margin with perturbation at only one layer, which is for any $l \in [L]$, we have

$$\begin{aligned} &m_F(\boldsymbol{x}_i, y_i) \leq \tilde{m}_{F,l}(\boldsymbol{x}_i, y_i) := \min_{\boldsymbol{\delta}_l} \|\boldsymbol{\delta}_l\|_2, \\ &\text{s.t. } \arg\max F(\boldsymbol{x}_i, \mathbf{0}, \dots, \boldsymbol{\delta}_l, \dots, \mathbf{0}) \neq y_i. \end{aligned} \tag{3}$$

Otherwise we find better minimizers for equation 1. The complete proof is provided in Appendix B. This offers us possibilities to investigate the input margin for every layer independently.

The minimal perturbation for each layer that alters the prediction of the model requires complete knowledge about the decision boundary around the sample, which is hard to find. However, with the classification loss function ℓ, we can find the steepest direction towards loss increment, approximating the direction towards misclassification for a sample. Therefore, we approximate the perturbation direction with the gradient of the layer with respect to its input, that is

$$\begin{aligned} \tilde{m}_{F,l}(\boldsymbol{x}_i, y_i) &= \min_{\boldsymbol{\delta}_l} \|\boldsymbol{\delta}_l\|_2 \\ &\approx \min_{\alpha_{i,l}} \|\alpha_{i,l} \nabla_{\boldsymbol{z}} \ell\left(F_l(\boldsymbol{z}_{i,l})\right)\|_2, \\ \text{s.t. } \arg\max F(\boldsymbol{x}_i, \mathbf{0}, \dots, \boldsymbol{\delta}_l, \dots, \mathbf{0}) &\neq y_i, \end{aligned} \tag{4}$$

where $\alpha_{i,l}$ is a optimizable variable to control the magnitude of the gradient, $\boldsymbol{z}_{i,l}$ is the intermediate feature of $\boldsymbol{x}_i$, the input of f_l, $F_l(\boldsymbol{z}) = f_L \circ \cdots \circ f_l(\boldsymbol{z})$, and ℓ is some classification loss function, in our case, the cross entropy loss. Denoting the minimizer for equation 4 as $\alpha^*_{i,l}$, we have

$$\tilde{m}_{F,l}(\boldsymbol{x}_i, y_i) \approx \alpha^*_{i,l} \|\nabla_{\boldsymbol{z}} \ell\left(F_l(\boldsymbol{z}_{i,l})\right)\|_2. \tag{5}$$

So far, we have got the approximation of the margin for sample i and layer l. For the average margin on rehearsal samples, we set $\alpha_{i,l}$ to a fixed value for each sample, and count the number of samples that alter their predictions, which is the accuracy under the gradient magnitude of $\alpha_{i,l}$. Since the accuracy drop reflects the number of samples that

alter their predictions due to the increasing magnitude of the perturbation, the rate of decline of the accuracy is an estimator of the average margin on rehearsal samples. With lower rate of decline, less samples alter their prediction for the same magnitude of perturbation, which means larger expected margin, and vice versa.

Now we can investigate the evolution of margin in rehearsal-based CIL. We perform experiments on the networks trained by iCaRL (Rebuffi et al., 2017) on CIFAR100 for 10 evenly split tasks with 2000 maximum number of rehearsal samples. We investigate the accuracy change on the rehearsal samples from the initial task when applying different scales of the perturbation for the first three residual blocks of ResNet32 (He et al., 2016). We plot the accuracy on these samples with different perturbation scale α_l, for each model after training each task.

The results are shown in Figure 1. As we can see from the plots, the accuracy of the rehearsal samples drops monotonically when the scale of the perturbation is increasing. Note that *Upper Bound* in each plot is the accuracy drop on the samples from the current task. We find that this curve is roughly the same for each model, so there are few overlapped dotted lines in each figure. Comparing the curve between rehearsal samples and samples from the current task, we find that rehearsal samples show more higher rate of accuracy decline, indicating the margin of the rehearsal samples are much smaller than that of the samples form the current task. More importantly, samples in the rehearsal memory show more rapid accuracy decline with the same amount of perturbation scale as the task goes on. And it happens for every block of the network, which means the margin of the rehearsal samples is averagely smaller when they run through more training tasks. More plots with different settings can be found in Appendix C.

4.2. Decision Boundary Perspective

According to equation 3, the margin requires to find the minimal perturbation that alters the prediction of the sample, which is closely related to the decision boundary. We plot the decision boundary between the rehearsal samples and the samples from the current task by plotting the predictions of the interpolated samples between three samples (Somepalli et al., 2022). The interpolated samples form a grid on the plane of these three samples. Therefore, we can assign color to each sample according to the model's prediction on the sample. The results are shown in Figure 2. From the plots we can easily find that the samples from the current task occupies a lot more decision space than the rehearsal samples. Also, visually, there are regions with much smaller volume at the decision boundary around the rehearsal samples, reflecting the volatility around the decision boundary. For the evolution in CIL, as the task goes on, the decision boundary shrinks towards the sample from the rehearsal memory, which means it can no longer resist the same amount of perturbation, leading to smaller margin for rehearsal samples. Therefore, we reveal the margin shrinkage from the perspective of visualizations of decision boundary.

Algorithm 1 Multi-layer Rehearsal Feature Augmentation

Input: batch $\mathcal{B}$, current model $F_t(\boldsymbol{x})$
for $\boldsymbol{x}_i, y_i \in \mathcal{B}$ **do**
  if $\boldsymbol{x}_i, y_i \in \mathcal{M}$ **then**
    Sample $l \sim \mathcal{U}\{1, L\}$, $\hat{\beta} \sim \mathcal{U}(0, \beta)$
    $\mathcal{L}_{\text{cls},i} = \text{AugmentedForward}(F_t(\boldsymbol{x}), \boldsymbol{x}_i, y_i, l, \hat{\beta})$
  else
    $\mathcal{L}_{\text{cls},i} = \ell(F_t(\boldsymbol{x}_i), y_i)$
  end if
end for
$\mathcal{L}_{\text{cls}} = \frac{1}{|\mathcal{B}|} \sum_i \mathcal{L}_{\text{cls},i}$
Output: $\mathcal{L}_{\text{cls}}$
function AugmentedForward($F(\boldsymbol{x})$, $\boldsymbol{x}$, y, l, $\hat{\beta}$)
  $\boldsymbol{z}_l = f_{l-1} \circ \cdots \circ f_1(\boldsymbol{x})$
  $\hat{\boldsymbol{z}}_l = \boldsymbol{z}_l + \hat{\beta} \|\boldsymbol{z}_l\|_2 \nabla_{\boldsymbol{z}} \ell(F_l(\boldsymbol{z}_l), y)$ {Eq. 6}
  return $\ell(F_l(\hat{\boldsymbol{z}}_l), y)$ {Eq. 7}
end function

4.3. Multi-layer Rehearsal Feature Augmentation

In order to get better generalization for old tasks, we seek for acquiring larger all-layer margin on rehearsal samples. Therefore, we propose to augment the rehearsal samples at each block's input to generate competitive augmented features for training, which is Multi-layer Rehearsal Feature Augmentation (MRFA). These augmented samples push the rehearsal samples away from the decision boundary and offer robustness against perturbations, leading to larger margin thus better generalization on old tasks.

Rehearsal-based methods often train incremental tasks together with the rehearsal memory. Each batch is comprised of both samples from current task and rehearsal memory. To augment the rehearsal samples, for each batch, we first compute the gradients of the model with respect to each block's input as the direction of the augmentation. Then for each rehearsal sample in the batch, we uniformly select one of the blocks' inputs to apply the augmentation, and compute the loss with augmented features, update the model parameters according to the augmented loss. Specifically, for each rehearsal sample $\boldsymbol{x}_i$ in the batch, we do the following computations:

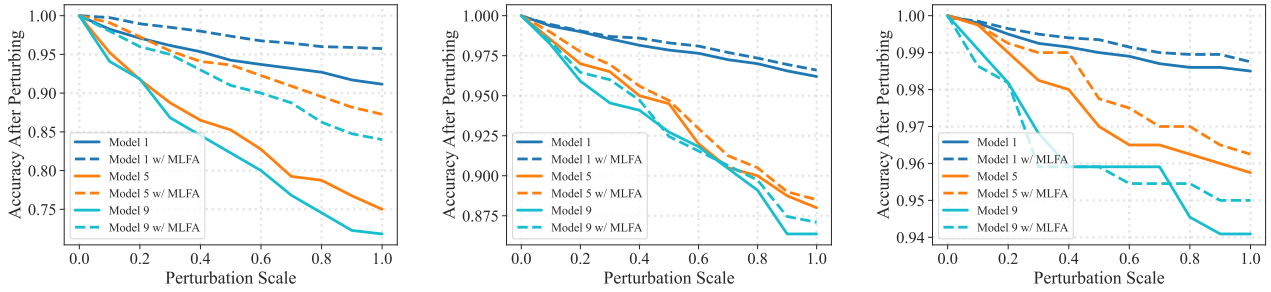
$$\hat{\boldsymbol{z}}_{i,l} = \boldsymbol{z}_{i,l} + \lambda \nabla_{\boldsymbol{z}} \ell(F_l(\boldsymbol{z}_{i,l})), \tag{6}$$

$$\mathcal{L}_{\text{cls},i} = \ell(F_l(\hat{\boldsymbol{z}}_{i,l}), y_i), \tag{7}$$

where $l \sim \mathcal{U}\{1, L\}$, $\boldsymbol{z}_{i,l} = f_{l-1} \circ \cdots \circ f_1(\boldsymbol{x}_i)$ is the intermediate feature for $\boldsymbol{x}_i$, λ is the magnitude of the augmentation.

Table 1. Performance Experiment Results on CIFAR100. Bold font represents our method improves the baseline in this scenario.

Memory Size	500				1000				2000			
Scenarios	10-10		50-10		10-10		50-10		10-10		50-10	
	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg
Replay	30.50	50.83	30.29	41.66	38.55	56.65	38.33	47.72	45.57	61.95	45.80	54.63
w/ MRFA	**31.69**	**51.61**	**31.98**	**42.85**	**39.42**	**57.12**	**39.78**	**48.54**	**46.85**	**62.59**	**47.24**	**55.51**
iCaRL	32.11	53.24	36.16	50.59	41.50	59.98	44.79	56.23	48.65	64.52	50.56	60.08
w/ MRFA	**33.51**	**54.84**	**37.89**	**51.48**	**42.84**	**60.82**	**46.02**	**57.96**	**49.73**	**65.17**	**52.49**	**61.50**
FOSTER	41.54	63.15	48.98	60.32	56.06	71.55	51.40	61.91	62.20	74.49	59.80	67.54
w/ MRFA	**42.12**	**63.90**	**49.51**	**60.83**	**56.76**	**71.94**	**52.06**	**62.34**	**63.41**	**75.23**	**60.74**	**68.02**
DyTox+	52.61	69.29	53.16	65.97	58.47	73.48	56.29	66.71	62.06	75.54	66.75	73.36
w/ MRFA	**54.31**	**70.56**	**54.03**	**66.82**	**59.38**	**74.17**	**57.96**	**67.56**	**63.80**	**76.23**	**68.21**	**74.73**



(a) Block 1 (b) Block 2 (c) Block 3

Figure 3. Accuracy drop after different scales of perturbation with MRFA. Dashed curves represent the models trained with MRFA. With MRFA, the accuracy decline is slower with the same amount of perturbation scale. This indicates the margin of the rehearsal samples is averagely larger with our method.

For the value of λ, we use random scaled L2 norm of $\boldsymbol{z}_{i,l}$, which is $\lambda = \hat{\beta}\|\boldsymbol{z}_{i,l}\|_2$. Introducing the L2 norm in the augmentation intuitively balances the different scales for each block. Random scaled L2 norm diversifies the augmentation with different levels of difficulty, especially when the model is approaching convergence. The scale factor is drawn from a uniform distribution $\hat{\beta} \sim \mathcal{U}(0, \beta)$, where β is a hyperparameter, to avoid adversarial overfitting (Yu et al., 2022). Equation 6 performs the gradient ascent step to the selected layer l, and augments the input feature $\boldsymbol{z}_{i,l}$ with the magnitude of λ. Equation 7 calculates the augmented loss for the actual backward process. The pseudocode for MRFA is presented in Algorithm 1.

Distillation Compatibility. Knowledge distillation (Zhou et al., 2003; Zhou & Jiang, 2004; Hinton et al., 2015) is a well-known technique in transfer learning, and widely used in CIL. Some of the methods in CIL store the model trained on previous tasks for distillation (Li & Hoiem, 2017; Rebuffi et al., 2017; Douillard et al., 2022). This model is often called as teacher model, it provides knowledge about previous tasks in order to prevent the forgetting in current task training. The classification loss with MRFA is based on the augmented features of the current model. To make the best use of the teacher model with our augmentations, we apply exactly the same configuration of augmentation as current model in the teacher model for each rehearsal sample in the batch. We can do this easily because the teacher model shares the same architecture with the current model. The augmentation can be easily added to the layer-wise features in the teacher model.

5. Experiments

5.1. Experiment Settings

Datasets. Following most of the image classification benchmarks in CIL (Rebuffi et al., 2017; Wu et al., 2019), we use CIFAR100 and ImageNet100 to train the model incrementally. CIFAR100 (Krizhevsky, 2009) has 50,000 training and 10,000 testing samples with 100 classes in total. Each sample is a tiny image in 32×32 pixels. ImageNet100 (Deng et al., 2009) has 1,300 training samples and 50 test samples for each class.

Data Split. There are two common types of splits in CIL. The *small base* one equally divides all of the classes in a dataset (Rebuffi et al., 2017). The *large base* one uses half of the classes in a dataset as the base task (task 0), and

Table 2. Performance Experiment Results on ImageNet100. Bold font represents our method improves the baseline in this scenario.

Memory Size	500				1000				2000			
Scenarios	10-10		50-10		10-10		50-10		10-10		50-10	
	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg
Replay	34.46	54.82	36.68	47.27	45.06	61.79	46.00	53.28	51.10	66.00	52.02	58.33
w/ MRFA	**36.15**	**56.02**	**38.32**	**48.41**	**46.78**	**62.47**	**47.83**	**54.51**	**52.51**	**67.22**	**53.61**	**59.54**
iCaRL	35.90	57.11	38.14	54.83	45.98	62.59	47.50	59.59	50.90	66.83	54.22	64.08
w/ MRFA	**37.24**	**58.14**	**39.88**	**55.35**	**47.15**	**63.62**	**48.89**	**60.24**	**52.05**	**67.34**	**55.78**	**65.15**
FOSTER	37.62	60.25	60.82	73.54	52.64	68.50	64.26	74.60	65.68	76.74	71.60	77.37
w/ MRFA	**39.45**	**61.57**	**62.03**	**74.36**	**53.87**	**69.77**	**65.93**	**75.44**	**66.93**	**77.61**	**72.88**	**78.49**
DyTox+	56.22	72.84	61.98	74.48	60.41	74.69	65.24	76.25	65.78	76.35	71.32	78.08
w/ MRFA	**57.54**	**73.38**	**62.46**	**75.25**	**61.84**	**75.57**	**65.99**	**76.68**	**66.56**	**77.31**	**72.57**	**79.29**

(a) Input Space of Model 1 (b) Input Space of Model 5 (c) Input Space of Model 9

(d) Input Space of Model 1 (e) Input Space of Model 5 (f) Input Space of Model 9

Figure 4. Evolution of decision boundary around rehearsal samples with MRFA. **Compared to Figure 2, the decision boundary around the rehearsal samples shrinks slower with our method.**

equally divides the remaining classes (Hou et al., 2019; Yu et al., 2020). For a dataset with 100 classes, 10-10 means 10 classes in the base task and all of the following incremental tasks are also with 10 classes, 50-10 means 50 classes in the base task and 10 classes in the incremental tasks.

Backbones and Baselines. In this paper, we test our method on four baselines. *Replay* is the baseline method with only the fixed number of rehearsal samples and no other techniques. We use ResNet32 (He et al., 2016) for CIFAR100 and ResNet18 for ImageNet100 as the backbone in Replay. iCaRL (Rebuffi et al., 2017) is a classic non-expanding method for CIL, which also uses ResNet32 and ResNet18 as the backbones for CIFAR100 and ImageNet100 respectively. FOSTER (Wang et al., 2022a) is based on feature boosting, which boosts the final representation for prediction. Dytox+ (Douillard et al., 2022) is a ViT-based method with expanding task tokens, which uses Convit (d'Ascoli et al., 2021) as the backbone.

Implementation. The experiments based on ResNet are implemented with the open-source code PyCIL (Zhou et al., 2023a). The experiments based on ViT-based backbones are implemented with the open-source code of DyTox (Douillard et al., 2022). The code is available on GitHub[1].

5.2. Performance Experiments

To verify the effectiveness of our proposed methods in CIL, we test MRFA on four baselines, covering both convolutional and transformer-based backbones, also covering both non-expandable and expandable architectures. The results for CIFAR100 and ImageNet100 are shown in Table 1 and Table 2. The results show that MRFA successfully improves rehearsal-based CIL in most of the scenarios by 1∼2%. This indicates that our method effectively alleviates the

[1] https://github.com/bwnzheng/MRFA_ICML2024

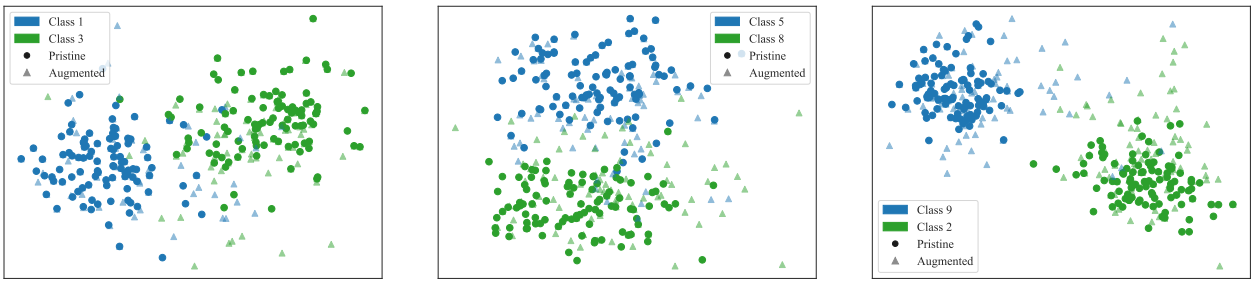


Figure 5. Visualizations of feature space augmented by MRFA. From the figures, we can conclude that the features augmented by MRFA adaptively fill the feature space around rehearsal samples, helping the model to attain refined decision boundaries.

Table 3. Average accuracy of MRFA with different selection strategies of rehearsal samples.

Memory Size	500		2000	
Scenarios	10-10	50-10	10-10	50-10
Replay w/ herding	54.82	47.27	61.95	54.63
w/ MRFA	56.02	48.41	62.59	55.51
Replay w/ random	54.20	46.61	61.29	54.05
w/ MRFA	55.48	47.86	62.17	54.95

Table 4. Average accuracy of MRFA on both current task and rehearsal samples (MRFA-a).

Memory Size	500		2000	
Scenarios	10-10	50-10	10-10	50-10
Replay w/ MRFA	51.61	42.85	62.59	55.51
Replay w/ MRFA-a	50.16	41.03	61.54	54.28
iCaRL w/ MRFA	54.84	51.48	65.17	61.50
iCaRL w/ MRFA-a	52.31	49.85	64.23	59.82

overfitting on rehearsal samples. We also record the training duration of the methods with MRFA, which shows that MRFA improves the baselines without additional samples with small overhead. Details are in section 5.6.

5.3. Margin Evolutions and Decision Boundaries

To verify the effectiveness of MRFA on making the margin larger, we plot the accuracy drop after different scales of perturbation when applying MRFA in CIL, just like we have done in section 4.1. The results are shown in Figure 3. As we can see from the figure, MRFA actually makes the rate of accuracy decline slower, indicating a larger upper bound on the margin of the rehearsal samples.

Also, we plot the evolution of the decision boundary around rehearsal samples when applying MRFA, just like we have done in section 4.1. The results are shown in Figure 4. As we can see from the figure, compared to Figure 2, MRFA effectively alleviates the shrinkage of the decision boundary, also indicating that a larger margin is acquired for the rehearsal samples.

5.4. Feature Space Visualization

In addition to the visualizations of the decision boundary, which focuses on the input space of the model, we also investigate the augmented feature in the feature space. To show the properties of the augmented features, we plot the final features of the rehearsal samples before and after the augmentation, using Principle Component Analysis. For the tidiness and clarity, we only show two classes in one plot. The results are shown in Figure 5. The magnitude of the augmentation is exaggerated for better illustration of the augmentation direction. As we can see from the figure, the augmented features are closer to other classes, which means MRFA adaptively augments the feature towards the decision boundary. The augmented features would fill the feature space around the rehearsal samples and help the model to attain refined decision boundaries.

5.5. Reducing Computational Cost

According to Alg. 1, the MRFA requires an extra forward-backward pass before each batch. However, it introduces large computational cost during the training of each task. There is a simple way to reduce such computational complexity, which is to perform such a pass for every p batch. To find out the effectiveness of such reduction, we perform experiments on CIFAR100 50-10, with different values of p, showing the average incremental accuracy (Avg) and total training duration in seconds. The results are shown in Figure 6. As we can see from the figure, the computational cost is reduced dramatically with such modification, with negligible performance loss.

5.6. Further Analysis

MRFA with different selection strategies of rehearsal samples. Herding (Welling, 2009) is a widely-adopted re-

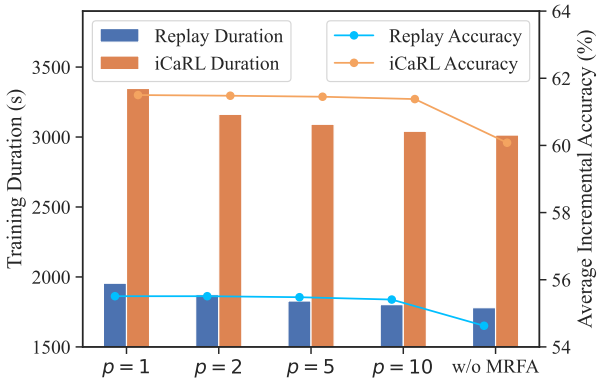


Figure 6. Training durations and accuracies of MRFA with different augmentation batch interval p.

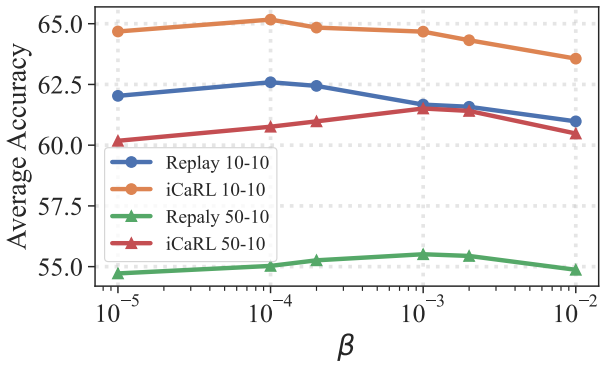


Figure 7. Average accuracy with different values of β.

hearsal selection strategy in CIL. It selects the samples greedily one by one to make the current selected samples approximate the prototype feature of the class. It is also the strategy we use in the performance experiments. We also test MRFA on different selection strategies of rehearsal samples on CIFAR100. The results are shown in Table 3. We perform MRFA with random selection and eviction for rehearsal samples. As the results show, MRFA also achieves performance improvement with random selection.

MRFA on both current task and rehearsal samples. The experiments above shows that larger margin on rehearsal samples indeed improves the generalizations on old tasks. But can we further improve the performance by applying MRFA on both current task and rehearsal samples (MRFA-a)? We find out the answer is no after performing experiments on CIFAR100 about this. As we can see in Table 4, MRFA-a performs generally worse than that only augments the rehearsal samples for each batch. This is caused by the imbalanced number of samples between rehearsal samples and samples form current task. Therefore, feature augmentation has stronger impact on the samples from current task, depriving more feature space occupied by old tasks, which leads to worse forgetting on old tasks.

Hyperparameter Sensitivity. MRFA requires one hyperparameter β which controls the maximum scale of the feature augmentations. We test the performance of MRFA for different values of β. The results are shown in Figure 7. As we can see from the figure, for 10-10, the best β is around 1e-4, while for 50-10, the best β is around 1e-3. The hyperparameter selections for each experiment are presented in Appendix D.

6. Conclusion

In this paper, we start from studying the overfitting on rehearsal samples in CIL. We reveal the overfitting on rehearsal samples from another perspective by studying how the all-layer margin of the network evolves in CIL. Specifically, we investigate the all-layer margin of the network quantitatively with its upper bound and show the decision boundary visually. As a result, we find that the margin shrinkage indeed happens in rehearsal samples. Therefore, we can alleviate the overfitting on the rehearsal samples by making the margin larger. Furthermore, we propose multi-layer rehearsal feature augmentation (MRFA) for class-incremental learning. The proposed method augments the features of rehearsal samples at the input of each block by gradient ascent step of the current model with respect to the feature. With such augmentation on block features, rehearsal samples are able to provide more information for refining the decision boundary during incremental learning, thus alleviating catastrophic forgetting. Moreover, extensive experiments are performed to show the effectiveness of MRFA on various settings of CIL.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

Acknowledgements

This work is partially supported by National Science and Technology Major Project (2022ZD0114805), NSFC (62376118, 62006112, 62250069, 61921006), Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

Boser, B. E., Guyon, I. M., and Vapnik, V. N. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152, 1992.

Buzzega, P., Boschini, M., Porrello, A., Abati, D., and Calderara, S. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.

Cha, S., Hsu, H., du Pin Calmon, F., , and Moon, T. CPR: Classifier-projection regularization for continual learning. In *4th Lifelong Machine Learning Workshop at ICML 2020*, 2020. URL https://openreview.net/forum?id=QZ2WhuvAE0B.

Chao, W.-L., Ye, H.-J., Zhan, D.-C., Campbell, M., and Weinberger, K. Q. Revisiting meta-learning as supervised learning. *arXiv preprint arXiv:2002.00573*, 2020.

Chaudhry, A., Rohrbach, M., Elhoseiny, M., Ajanthan, T., Dokania, P. K., Torr, P. H., and Ranzato, M. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE, 2009.

Douillard, A., Cord, M., Ollion, C., Robert, T., and Valle, E. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Proceedings of the European conference on computer vision*, pp. 86–102. Springer, 2020.

Douillard, A., Ramé, A., Couairon, G., and Cord, M. Dytox: Transformers for continual learning with dynamic token expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9285–9295, 2022.

Drucker, H., Burges, C. J., Kaufman, L., Smola, A., and Vapnik, V. Support vector regression machines. *Advances in neural information processing systems*, 9:155–161, 1996.

d'Ascoli, S., Touvron, H., Leavitt, M. L., Morcos, A. S., Biroli, G., and Sagun, L. Convit: Improving vision transformers with soft convolutional inductive biases. In *International Conference on Machine Learning*, pp. 2286–2296. PMLR, 2021.

Elsayed, G., Krishnan, D., Mobahi, H., Regan, K., and Bengio, S. Large margin deep networks for classification. *Advances in neural information processing systems*, 31: 842–852, 2018.

Golowich, N., Rakhlin, A., and Shamir, O. Size-independent sample complexity of neural networks. In *Conference On Learning Theory*, pp. 297–299. PMLR, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Hou, S., Pan, X., Loy, C. C., Wang, Z., and Lin, D. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 831–839, 2019.

Jia, M., Tang, L., Chen, B.-C., Cardie, C., Belongie, S., Hariharan, B., and Lim, S.-N. Visual prompt tuning. In *Proceedings of the European conference on computer vision*, pp. 709–727. Springer, 2022.

Jin, X., Sadhu, A., Du, J., and Ren, X. Gradient-based editing of memory examples for online task-free continual learning. *Advances in Neural Information Processing Systems*, 34:29193–29205, 2021.

Kakade, S. M., Sridharan, K., and Tewari, A. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. *Advances in neural information processing systems*, 21:793–800, 2008.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Koltchinskii, V. and Panchenko, D. Empirical margin distributions and bounding the generalization error of combined classifiers. *The Annals of Statistics*, 30(1):1–50, 2002.

Krizhevsky, A. Learning multiple layers of features from tiny images. 2009.

Kumari, L., Wang, S., Zhou, T., and Bilmes, J. A. Retrospective adversarial replay for continual learning. *Advances in Neural Information Processing Systems*, 35: 28530–28544, 2022.

Li, Z. and Hoiem, D. Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(12):2935–2947, 2017.

Lin, G., Chu, H., and Lai, H. Towards better plasticity-stability trade-off in incremental learning: A simple linear connector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 89–98, 2022.

Liu, Y., Su, Y., Liu, A.-A., Schiele, B., and Sun, Q. Mnemonics training: Multi-class incremental learning without forgetting. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pp. 12245–12254, 2020.

McCloskey, M. and Cohen, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychol. Learn. Motiv.*, volume 24, pp. 109–165. Elsevier, 1989.

Mirzadeh, S. I., Farajtabar, M., Gorur, D., Pascanu, R., and Ghasemzadeh, H. Linear mode connectivity in multitask and continual learning. In *International Conference on Learning Representations*, 2020.

Nagarajan, V. and Kolter, Z. Deterministic pac-bayesian generalization bounds for deep networks via generalizing noise-resilience. In *International Conference on Learning Representations*, 2019.

Neyshabur, B., Tomioka, R., and Srebro, N. Norm-based capacity control in neural networks. In *Conference on learning theory*, pp. 1376–1401. PMLR, 2015.

Ostapenko, O., Rodriguez, P., Caccia, M., and Charlin, L. Continual learning via local module composition. *Advances in Neural Information Processing Systems*, 34: 30298–30312, 2021.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

Robins, A. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2):123–146, 1995.

Shi, Y., Zhou, K., Liang, J., Jiang, Z., Feng, J., Torr, P. H., Bai, S., and Tan, V. Y. Mimicking the oracle: An initial phase decorrelation approach for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16722–16731, 2022.

Shin, H., Lee, J. K., Kim, J., and Kim, J. Continual learning with deep generative replay. *Advances in Neural Information Processing Systems*, 30:2990–2999, 2017.

Sokolić, J., Giryes, R., Sapiro, G., and Rodrigues, M. R. Robust large margin deep neural networks. *IEEE Transactions on Signal Processing*, 65(16):4265–4280, 2017.

Somepalli, G., Fowl, L., Bansal, A., Yeh-Chiang, P., Dar, Y., Baraniuk, R., Goldblum, M., and Goldstein, T. Can neural nets learn the same model twice? investigating reproducibility and double descent from the decision boundary perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13699–13708, 2022.

Tiwari, R., Killamsetty, K., Iyer, R., and Shenoy, P. Gcr: Gradient coreset based replay buffer selection for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 99–108, 2022.

Vapnik, V. *The nature of statistical learning theory*. Springer science & business media, 1999.

Verwimp, E., De Lange, M., and Tuytelaars, T. Rehearsal revealed: The limits and merits of revisiting samples in continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9385–9394, 2021.

Wang, F.-Y., Zhou, D.-W., Ye, H.-J., and Zhan, D.-C. Foster: Feature boosting and compression for class-incremental learning. In *Proceedings of the European conference on computer vision*, pp. 398–414. Springer, 2022a.

Wang, Y., Huang, Z., and Hong, X. S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning. *Advances in Neural Information Processing Systems*, 35:5682–5695, 2022b.

Wang, Z., Zhang, Z., Ebrahimi, S., Sun, R., Zhang, H., Lee, C.-Y., Ren, X., Su, G., Perot, V., Dy, J., et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. *European Conference on Computer Vision*, 2022c.

Wang, Z., Zhang, Z., Lee, C.-Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J., and Pfister, T. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 139–149, 2022d.

Wei, C. and Ma, T. Improved sample complexities for deep neural networks and robust classification via an all-layer margin. In *International Conference on Learning Representations*, 2020.

Welling, M. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1121–1128, 2009.

Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., and Fu, Y. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 374–382, 2019.

Yan, S., Xie, J., and He, X. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3014–3023, 2021.

Yan, Z., Guo, Y., and Zhang, C. Adversarial margin maximization networks. *IEEE transactions on pattern analysis and machine intelligence*, 43(4):1129–1139, 2019.

Yang, E., Shen, L., Wang, Z., Liu, S., Guo, G., and Wang, X. Data augmented flatness-aware gradient projection for continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5630–5639, 2023.

Yu, C., Han, B., Shen, L., Yu, J., Gong, C., Gong, M., and Liu, T. Understanding robust overfitting of adversarial training and beyond. In *International Conference on Machine Learning*, pp. 25595–25610. PMLR, 2022.

Yu, L., Twardowski, B., Liu, X., Herranz, L., Wang, K., Cheng, Y., Jui, S., and Weijer, J. v. d. Semantic drift compensation for class-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6982–6991, 2020.

Zhang, Y., Pfahringer, B., Frank, E., Bifet, A., Lim, N. J. S., and Jia, Y. A simple but strong baseline for online continual learning: Repeated augmented rehearsal. *Advances in Neural Information Processing Systems*, 35: 14771–14783, 2022.

Zhao, P., Zhang, Y.-J., Zhang, L., and Zhou, Z.-H. Adaptivity and non-stationarity: Problem-dependent dynamic regret for online convex optimization. *Journal of Machine Learning Research*, 25(98):1–52, 2024.

Zheng, B., Zhou, D.-W., Ye, H.-J., and Zhan, D.-C. Preserving locality in vision transformers for class incremental learning. In *2023 IEEE International Conference on Multimedia and Expo*, pp. 1157–1162, 2023.

Zhou, D.-W., Wang, F.-Y., Ye, H.-J., and Zhan, D.-C. Pycil: a python toolbox for class-incremental learning. *SCIENCE CHINA Information Sciences*, 66(9):197101–, 2023a.

Zhou, D.-W., Wang, Q.-W., Qi, Z.-H., Ye, H.-J., Zhan, D.-C., and Liu, Z. Deep class-incremental learning: A survey. *arXiv preprint arXiv:2302.03648*, 2023b.

Zhou, D.-W., Wang, Q.-W., Ye, H.-J., and Zhan, D.-C. A model or 603 exemplars: Towards memory-efficient class-incremental learning. In *The Eleventh International Conference on Learning Representations*, 2023c.

Zhou, D.-W., Ye, H.-J., Zhan, D.-C., and Liu, Z. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *arXiv:2303.07338*, 2023d. unpublished.

Zhou, Z.-H. Learnability with time-sharing computational resource concerns. *ArXiv preprint*, arXiv:2305.02217, 2023.

Zhou, Z.-H. and Jiang, Y. Nec4. 5: Neural ensemble based c4. 5. *IEEE Transactions on knowledge and data engineering*, 16(6):770–773, 2004.

Zhou, Z.-H., Jiang, Y., and Chen, S.-F. Extracting symbolic rules from trained neural network ensembles. *AI Communications*, 16:3–15, 2003.

Zhu, F., Cheng, Z., Zhang, X.-Y., and Liu, C.-l. Class-incremental learning via dual augmentation. *Advances in Neural Information Processing Systems*, 34:14306–14318, 2021.

A. Additional Related Works

In this section, we discuss additional related works in class-incremental learning.

Model expansion comes from the idea of parameter isolation for each task (Yan et al., 2021; Ostapenko et al., 2021; Douillard et al., 2022; Zhou et al., 2023c). It expands the representation space as the task goes on. DER (Yan et al., 2021) trains a separate backbone for each task, aggregating all of the representations for classification. DyTox (Douillard et al., 2022) learns a separate task token for each task.

Knowledge distillation (Li & Hoiem, 2017; Douillard et al., 2020) uses the model trained on previous tasks as a teacher and distillation losses to keep the previously learned knowledge in the representation. LwF (Li & Hoiem, 2017) proposes to use the response of the old model to guide the training of the new model's old tasks. PODNet (Douillard et al., 2020) uses the pooled intermediate feature maps of the ResNet to be the distillation target in training.

Regularization methods (Kirkpatrick et al., 2017; Zhu et al., 2021; Shi et al., 2022; Cha et al., 2020) come from various ideas. Kirkpatrick et al. (2017) proposes to restrict the updates of important parameters. Zhu et al. (2021) proposes a dual augmentation framework to make the eigenvalues of the representation's covariance matrix larger. Shi et al. (2022) proposes to make the representation scatter uniformly, making the representation contains more information about the input sample.

Pre-Trained Model-based methods leverage pretrained models and adapt the model for class-incremental learning (Wang et al., 2022d;b;c; Zhou et al., 2023d). Wang et al. (2022d) uses visual prompt tuning (Jia et al., 2022) to learn a prompt for each task. Wang et al. (2022c) proposes the dual prompt scheme in ViT. Due to the head start of the pre-trained models in learning representations, these methods outperform the methods which train the model from scratch, even without the rehearsal memory samples.

Other perspectives to boost CIL are also considered. (Zhu et al., 2021) proposes a dual augmentation framework to make the eigenvalues of the feature's covariance matrix larger. In the parameter space, (Mirzadeh et al., 2020) studies the linear mode connectivity in CIL and proposes to enhance the linear mode connectivity between learned models. (Lin et al., 2022) also considers the linear mode connectivity between learned models and proposes to combine two models learned in different ways to get better linear mode connectivity. (Zheng et al., 2023) proposes a locality-preserving attention module to remedy the locality degradation during the training of CIL.

B. Proof of Formula 3

Let the minimizers of all-layer margin in equation 1 be $\boldsymbol{\delta}_{l'}^*$ for $l' \in [L]$. Formally,

$$\begin{aligned} \boldsymbol{\delta}_1^*, \ldots, \boldsymbol{\delta}_L^* := \underset{\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_L}{\arg\min} \sqrt{\sum_{l=1}^{L} \|\boldsymbol{\delta}_l\|_2^2}, \\ \text{s.t.} \arg\max F(\boldsymbol{x}_i, \boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_L) \neq y_i. \end{aligned} \tag{8}$$

Let the minimizer for equation 3 be $\tilde{\boldsymbol{\delta}}_l^*$. Formally,

$$\begin{aligned} \tilde{\boldsymbol{\delta}}_l^* = \underset{\boldsymbol{\delta}_l}{\arg\min} \|\boldsymbol{\delta}_l\|_2, \\ \text{s.t.} \arg\max F(\boldsymbol{x}_i, 0, \ldots, \boldsymbol{\delta}_l, \ldots, 0) \neq y_i. \end{aligned} \tag{9}$$

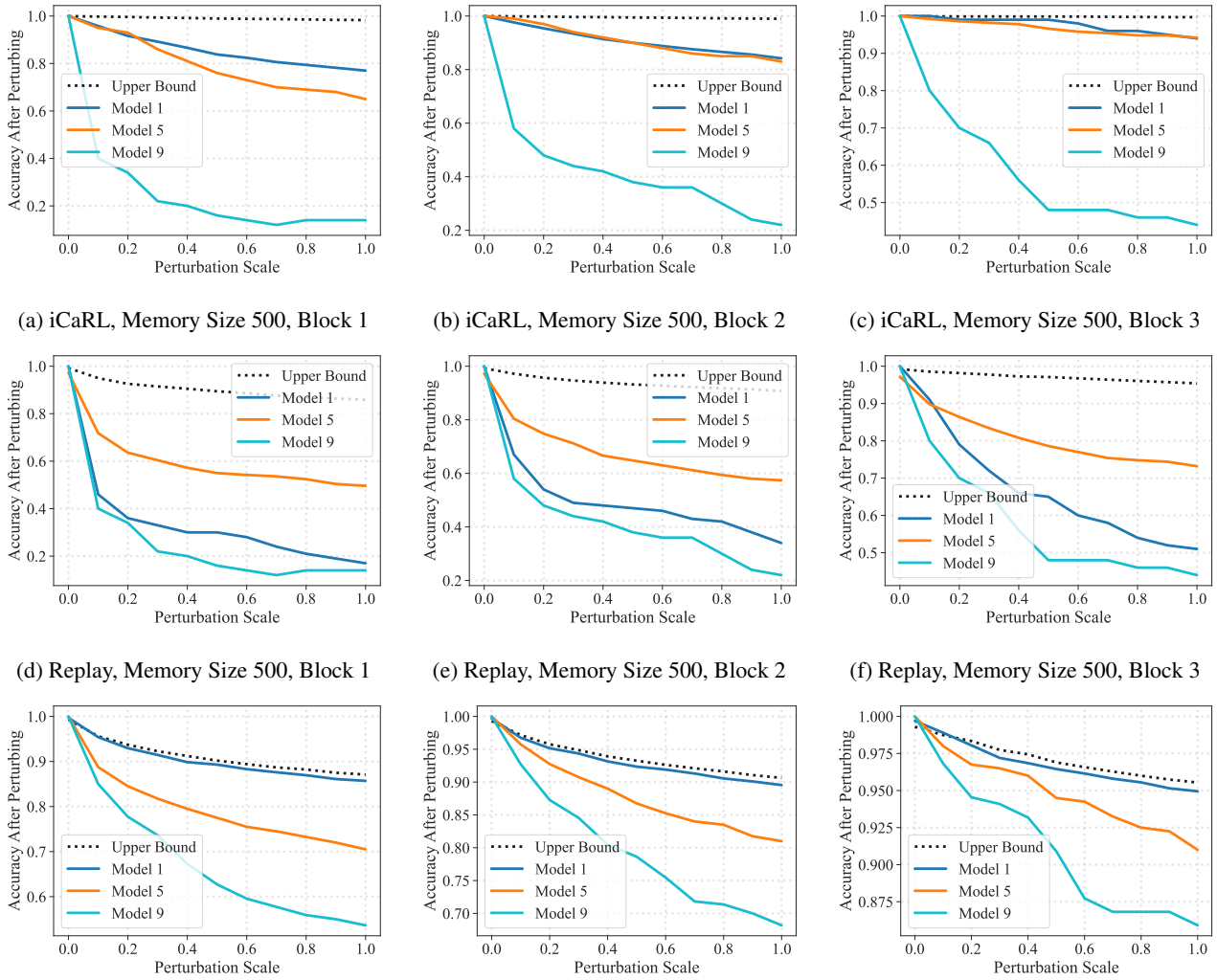
The objective is formulated as $\forall l \in [L]$,

$$\|\tilde{\boldsymbol{\delta}}_l^*\|_2 \geq \sqrt{\sum_{l'=1}^{L} \|\boldsymbol{\delta}_{l'}^*\|_2^2}. \tag{10}$$

We will prove it by contradiction. Suppose $\|\tilde{\boldsymbol{\delta}}_l^*\|_2 < \sqrt{\sum_{l'=1}^{L} \|\boldsymbol{\delta}_{l'}^*\|_2^2}$, we find better minimizers for equation 1. The better minimizers are $(0, \ldots, \tilde{\boldsymbol{\delta}}_l^*, \ldots, 0)$, they have lower cost than $(\boldsymbol{\delta}_1^*, \ldots, \boldsymbol{\delta}_L^*)$. Therefore, $\|\tilde{\boldsymbol{\delta}}_l^*\|_2 \geq \sqrt{\sum_{l'=1}^{L} \|\boldsymbol{\delta}_{l'}^*\|_2^2}$ holds for any $l \in [L]$.

C. More Empirical Analysis Results

We perform empirical analysis on accuracy drop after different scales of perturbation with various settings. The results are shown in Figure 8. We find similar results as in the main text from these plots.



(a) iCaRL, Memory Size 500, Block 1 (b) iCaRL, Memory Size 500, Block 2 (c) iCaRL, Memory Size 500, Block 3

(d) Replay, Memory Size 500, Block 1 (e) Replay, Memory Size 500, Block 2 (f) Replay, Memory Size 500, Block 3

(g) Replay, Memory Size 2000, Block 1 (h) Replay, Memory Size 2000, Block 2 (i) Replay, Memory Size 2000, Block 3

Figure 8. Accuracy drop after different scales of perturbation with various settings.

D. Hyperparameter Selections

The values of the hyperparameter β for each experiment in section 5.2 are listed in Table 5.

Table 5. Hyperparameter (β) selections for each experiment in section 5.2

Memory Size	500		1000		2000	
Scenarios	10-10	50-10	10-10	50-10	10-10	50-10
Replay w/ MRFA	1e-3	1e-3	1e-4	1e-3	1e-4	1e-3
iCaRL w/ MRFA	1e-3	1e-3	1e-4	1e-3	1e-4	1e-3
FOSTER w/ MRFA	1e-3	1e-3	1e-4	1e-3	1e-4	1e-3
DyTox+ w/ MRFA	1e-3	1e-3	1e-4	1e-3	1e-4	1e-3

E. More Implementation Details

We provide more implementation details for performance experiments in Table 6. We use the same configuration for CIFAR100 and ImageNet100 without additional specification.

Table 6. Training settings for baselines in section 5.2

Baselines	# of Base Epochs	# of Incremental Epochs	Batch Size	Learning Rate	# of GPUs
Replay	200	70	128	0.1	1
iCaRL	200	170	128	0.1	1
FOSTER	200	170	128	0.1	1
DyTox+	500	500	128	5e-4	2 (CIFAR100) 4 (ImageNet100)