# Learning a Unified Classifier Incrementally via Rebalancing

Saihui Hou[1*], Xinyu Pan[2*], Chen Change Loy[3], Zilei Wang[1], Dahua Lin[2]
[1] University of Science and Technology of China, [2] The Chinese University of Hong Kong,
[3] Nanyang Technological University
saihui@mail.ustc.edu.cn, px118@ie.cuhk.edu.hk, ccloy@ntu.edu.sg,
zlwang@ustc.edu.cn, dhlin@ie.cuhk.edu.hk

## Abstract

*Conventionally, deep neural networks are trained offline, relying on a large dataset prepared in advance. This paradigm is often challenged in real-world applications,* e.g. *online services that involve continuous streams of incoming data. Recently, incremental learning receives increasing attention, and is considered as a promising solution to the practical challenges mentioned above. However, it has been observed that incremental learning is subject to a fundamental difficulty – catastrophic forgetting, namely adapting a model to new data often results in severe performance degradation on previous tasks or classes. Our study reveals that the imbalance between previous and new data is a crucial cause to this problem. In this work, we develop a new framework for incrementally learning a unified classifier,* i.e. *a classifier that treats both old and new classes uniformly. Specifically, we incorporate three components, cosine normalization, less-forget constraint, and inter-class separation, to mitigate the adverse effects of the imbalance. Experiments show that the proposed method can effectively rebalance the training process, thus obtaining superior performance compared to the existing methods. On CIFAR-100 and ImageNet, our method can reduce the classification errors by more than 6% and 13% respectively, under the incremental setting of 10 phases.*

## 1. Introduction

Incremental learning is a learning paradigm that allows a model to be *continually* updated on new data, instead of being trained once on a whole dataset. In recent years, incremental learning sees increasing demand from real-world applications – many of them are exposed to continuous streams of data during daily operation. A natural approach to incremental learning is to simply finetune a pretrained model on new data. This approach, however, faces a serious challenge – *catastrophic forgetting* [26]. To be more specific, finetuning a model on new data usually results in significant performance drop on previous data.

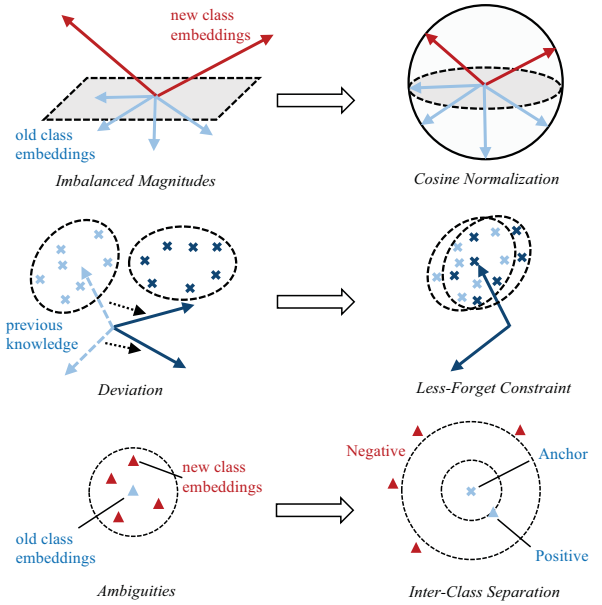



Figure 1. Illustration of the adverse effects caused by the *imbalance* between old and new classes in *multi-class* incremental learning, and how our approach tackle them.

Great efforts have been devoted to overcoming this difficulty, which generally follow two directions: (1) trying to identify and preserve significant parameters of the original model [21, 37, 1], and (2) trying to preserve the knowledge in the original model through methods like knowledge distillation [24, 2, 28, 29, 19, 15]. Whereas these methods, to a certain extent, mitigate the effect of catastrophic forgetting, the overall performances remain significantly inferior to those obtained by joint training.

In this work, we aim to explore a more effective way to incremental learning. Particularly, we focus on the *multi-class setting*, with an aim to learn a *unified* classifier that can

* indicates joint first authorship.

recognize all classes seen at different stages. Compared to the conventional *multi-task setting*, where a model is trained to handle different tasks with each task dedicated to a separate group of classes, the multi-class setting is more realistic, but also more challenging.

As we work on this problem, we find that the *imbalance* between the old classes seen at previous stages and the new ones at the current stage constitutes a key challenge. Specifically, the training algorithm only sees none or a few samples of old classes but substantially more of new classes. Under this circumstance, the focus of the training process is significantly biased towards new classes, thus leading to a number of adverse effects on the class-specific weights as shown in Figure 1: (1) *imbalanced magnitudes:* the magnitudes of the weight vectors of new classes are remarkably higher than those of old classes; (2) *deviation:* the previous knowledge, *i.e*. the relationship between the features and the weight vectors of old classes, are not well preserved; and (3) *ambiguities:* the weight vectors of new classes are close to those of old classes, often leading to ambiguities. The combination of these effects can severely mislead the classifier, resulting in the decisions biased towards new classes and the confusion among old classes.

In response to these problems, we propose a new framework for learning a unified classifier under the incremental setting. Particularly, it incorporates three components to mitigate the adverse effects caused by the imbalance: (1) *cosine normalization*, which enforces balanced magnitudes across all classes, including both old and new ones; (2) *less-forget constraint*, which aims to preserve the geometric configuration of old classes; and (3) *inter-class separation*, which encourages a large margin to separate the old and new classes. By rebalancing the training process with these techniques, the proposed framework can more effectively preserve the knowledge learned in previous phases and reduce the ambiguities between old and new classes.

We systematically compare different methods for incremental learning on CIFAR-100 [22] and ImageNet [7], under the *multi-class* setting. In our experiments, the proposed framework performs significantly superior to the baselines. For example, under the incremental settings of 10 phases on CIFAR100 and ImageNet, our method can reduce the classification errors by more than 6% and 13% respectively.

## 2. Related Work

### 2.1. Incremental Learning

Incremental learning has been a long standing research area [4, 30]. Recently, along with the success of deep learning, incremental learning of deep neural networks becomes an active topic, where the existing works mainly fall into two categories, *parameter-based* and *distillation-based*.

**Parameter-based.** The methods of this category such as EWC [21], SI [37], MAS [1] try to estimate the importance of each parameter in the original model and add more penalty to the changes on significant parameters. The differences among these works lie in the way to compute the parameter importance. However, it is difficult to design a reasonable metric to evaluate all the parameters, especially in long sequences of tasks or classes.

**Distillation-based.** Knowledge distillation, as discussed in [14], is an effective way to transfer knowledge from one network to another. It is first introduced to incremental learning in *Learning without Forgetting (LwF)* [24], where a modified cross-entropy loss is used to preserve the knowledge in the original model. Aljundi *et al*. [2] propose to train multiple networks on different tasks and take an auto-encoder to choose one for each test sample. Rannen *et al*. [28] also introduce an auto-encoder to preserve the crucial features for old tasks. Hou *et al*. [15] propose to use knowledge distillation to facilitate the adaptation to new tasks. Note that the works mentioned above [24, 2, 28, 15] all follow the *multi-task* setting, *i.e*. the trained model is equipped with multiple classifiers, each of which is evaluated only on the data from an individual task.

The *multi-class* setting, which aims to learn a unified classifier for all the classes observed so far, has also been explored in previous efforts [19, 29, 3]. Jung *et al*. [19] consider the domain expansion that can be treated as a special case of incremental learning, and propose a solution that relies on two properties, namely unchanged decision boundaries and feature proximity. iCaRL [29] combines knowledge distillation and representation learning, with several novel components, *e.g*. *nearest-mean-of-exemplars* classification, and prioritized exemplar selection. Castro *et al*. [3] resort to the sophisticated data augmentation on the reserved old samples, reporting even better performance.

**Discussion.** In this work, the proposed method falls into the *distillation-based* category. But it differs from previous works in a key aspect: more than simply combining different objective terms to balance old and new classes, we carefully investigate the adverse effects of imbalance and propose a systematic solution that overcomes the issue from multiple perspectives.

It is noteworthy that the previous works have also explored other ideas for incremental learning, such as adopting dynamic network structures [31, 36] or using a generative model to produce samples for old classes [35, 20]. These works, however, are orthogonal to the proposed method, and thus can be incorporated into our framework to achieve further improvement.

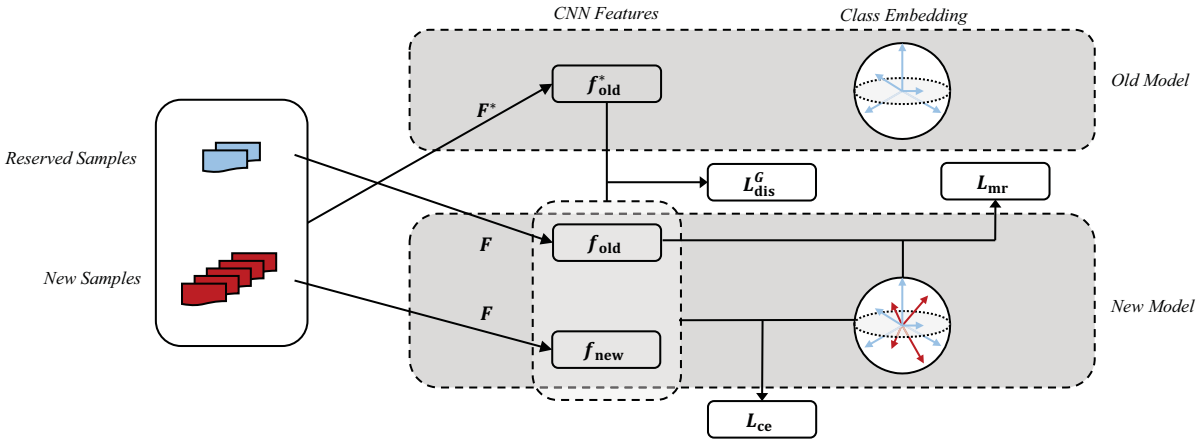



Figure 2. Illustration of our approach for *multi-class* incremental learning. Due to *cosine normalization*, the features and class embeddings lie in a high-dimensional sphere geometrically. There are three types of loss involved in the incremental process. Besides the cross-entropy loss $L_{\text{ce}}$ computed on all classes, $L^{\text{G}}_{\text{dis}}$ is a novel distillation loss computed on the features (*less-forget constraint*), and $L_{\text{mr}}$ is a variant of margin ranking loss to separate the old and new classes (*inter-class separation*).

### 2.2. Tackling Imbalance

Class imbalance is a significant challenge for machine learning [18, 13]. Previous efforts to tackle the class imbalance can be roughly divided into two groups: data resampling [13, 6, 12] and cost-sensitive learning [32, 17, 38, 16, 9]. The former aims to rebalance the training samples in different classes through resampling; while the latter focuses on adjusting the loss. In this work, we tackle the imbalance in incremental learning from different aspects, instead of directly adjusting the sampling ratio or loss weights. In *inter-class separation*, we introduce a margin ranking loss which focuses on the boundary and thus is less susceptible to the imbalance among classes. Among the previous works, the one presented in [9] is the most relevant to ours. Dong *et al*. [9] propose a class rectification loss to rectify the learning bias of cross-entropy loss given the imbalanced data. Our margin ranking loss differs from [9] in the mining of the positives and hard negatives which is more efficient and specialized for incremental learning. In particular, we do not rely on a pretrained model to define the class similarity for the negative selection.

## 3. Our Approach

In this work we focus on the *multi-class* incremental classification problem. Formally, given a model trained on an old dataset $\mathcal{X}_{\text{o}}$, we aim to learn a unified classifier for both old classes $\mathcal{C}_{\text{o}}$ and new classes $\mathcal{C}_{\text{n}}$, based on a new dataset $\mathcal{X} = \mathcal{X}_{\text{n}} \cup \mathcal{X}'_{\text{o}}$. $\mathcal{X}_{\text{n}}$ is a large dataset that covers only the new classes $\mathcal{C}_{\text{n}}$, while $\mathcal{X}'_{\text{o}} \subset \mathcal{X}_{\text{o}}$ reserves just a tiny subset of old samples. The main challenge is how to utilize the severely imbalanced $\mathcal{X}$ and the original model to boost the performance on all classes without suffering from *catastrophic forgetting* [26]. In what follows, we will first review *Learning without forgetting (LwF)* [24] and iCaRL [29] as background. Then we will dive deeply into the imbalance in *multi-class* incremental learning, and elaborate on how our approach can address the issue from different aspects. The proposed approach is shown in Figure 2.

### 3.1. Background

*LwF* is the first work to introduce knowledge distillation to *multi-task* incremental learning and here we adapt it to the *multi-class* setting. For each training sample $x$, the loss function is the sum of two terms: the *classification loss* $L_{\text{ce}}$ and the *distillation loss* $L^{\text{F}}_{\text{dis}}$. Specifically, $L_{\text{ce}}$ is the standard cross-entropy loss [23]:

$$L_{\text{ce}}(x) = -\sum_{i=1}^{|\mathcal{C}|} y_i \log\left(p_i\right), \tag{1}$$

where $\mathcal{C}$ is the set of all observed classes so far, $y$ is the one-hot ground-truth label and $p$ is the corresponding class probabilities obtained by softmax. $L^{\text{F}}_{\text{dis}}$ is the distillation loss, which aims to make the current model mimic the behaviors of the original model, *i.e*. the model learned on old classes:

$$L^{\text{F}}_{\text{dis}}(x) = -\sum_{i=1}^{|\mathcal{C}_{\text{o}}|} \tau_i(p^*) \log(\tau_i(p)), \tag{2}$$

where $p^*$ is the soft label of $x$ generated by the original model on old classes, $\tau_i(v) = v_i^{1/\Omega} / \sum_j v_j^{1/\Omega}$ is a rescaling function, where $\Omega$ is usually set to be greater than 1 (*e.g*. $\Omega = 2$ in our experiments) to increase the weights of small values. While $L^{\text{F}}_{\text{dis}}$ is devised to preserve the previous knowledge by encouraging the current predictions on


Figure 3. Visualization of the weights and biases in the last layer for old and new classes. The results come from the incremental setting of CIFAR100 (1 phase) by iCaRL [29].

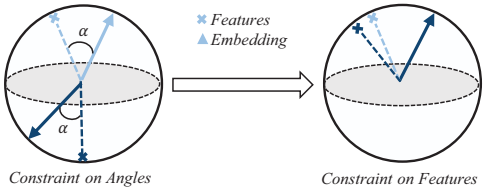



Figure 4. Illutration of *less-forget constraint*. The constraint on features is stronger compared to the constraint on angles with the embeddings of old classes fixed.

old classes to match the soft labels by the original model. However, it is observed in both our study and [29] that the adapted *LwF* tends to classify the test samples into new classes.

To deal with the issue, iCaRL [29] proposes a classification strategy named *nearest-mean-of-exemplars*. Specifically, it computes a prototype $\mu_i$ by averaging features of all reserved samples for each class $c_i \in \mathcal{C}$. During inference, it extracts the features for a test sample and assigns the class label of the most similar prototype. While iCaRL makes improvements over *LwF*, its performance on long sequences of classes is still not satisfying[1].

Overall, despite all the efforts devoted to incremental learning, there remains much room to improve. A key problem that limits the performance of the *multi-class* setting, as discussed earlier, is the significant imbalance between old and new classes. In this work, we aim to tackle this problem by incorporating three components, *cosine normalization*, *less-forget constraint*, and *inter-class separation*, which addresses the imbalance from different aspects. In what follows, we will present these components in turn.

### 3.2. Cosine Normalization

In a typical CNN, the predicted probability of a sample $x$ is computed as follows:

$$p_i(x) = \frac{\exp(\theta_i^{\mathrm{T}} f(x) + b_i)}{\sum_j \exp(\theta_j^{\mathrm{T}} f(x) + b_j)}, \tag{3}$$

where $f$ is the feature extractor, $\theta$ and $b$ are the weights (*i.e.* class embedding) and the bias vectors in the last layer. As shown in Figure 3, due to the class imbalance, the magnitudes of both the embeddings and the biases for the new classes are significantly higher than those for the old classes. This results in the *bias* in the predictions that favor new classes. To address this issue, we propose to use *cosine normalization* in the last layer, as:

$$p_i(x) = \frac{\exp(\eta \langle \bar{\theta}_i, \bar{f}(x) \rangle)}{\sum_j \exp(\eta \langle \bar{\theta}_j, \bar{f}(x) \rangle)}, \tag{4}$$

[1]The implementation of iCaRL described here is a little different from the original version [29]. Our implementation refers to those in [3, 35] which have proven to be more effective.

where $\bar{v} = v/\|v\|_2$ denotes the $l_2$-normalized vector, and $\langle \bar{v}_1, \bar{v}_2 \rangle = \bar{v}_1^{\mathrm{T}} \bar{v}_2$ measures the cosine similarity between two normalized vectors. The learnable scalar $\eta$ is introduced to control the peakiness of softmax distribution since the range of $\langle \bar{v}_1, \bar{v}_2 \rangle$ is restricted to $[-1, 1]$. Although *cosine normalization* is widely adopted in other visual tasks [33, 10, 27, 25], it is first introduced for incremental learning here. It can effectively eliminate the *bias* caused by the significant difference in magnitudes.

Now we revisit the method in Section 3.1 for incremental learning based on *cosine normalization*. For a sample $x$, the classification loss $L_{\mathrm{ce}}$ is computed similarly as in Eq (1) except that the probability for each class is computed in a different way. For the distillation loss, since the scalar $\eta$ in the original model and that in the current network are different, it is reasonable to mimic the scores before softmax instead of the probabilities after softmax. It is also noteworthy that due to *cosine normalization*, the scores before softmax all lies in the same range (*i.e.* $[-1, 1]$) and thus are comparable. Formally, the distillation loss is updated as:

$$L_{\mathrm{dis}}^{\mathrm{C}}(x) = -\sum_{i=1}^{|\mathcal{C}_{\mathrm{o}}|} \|\langle \bar{\theta}_i, \bar{f}(x) \rangle - \langle \bar{\theta}_i^*, \bar{f}^*(x) \rangle\|, \tag{5}$$

where $f^*$ and $\theta^*$ are the feature extractors and class embeddings in the original model, $|\mathcal{C}_{\mathrm{o}}|$ are the number of old classes. Geometrically, the normalized features and the class embeddings lie on a high-dimensional sphere. $L_{\mathrm{dis}}^{\mathrm{C}}$ encourages the geometric structures, reflected by the angles between the features and the old class embeddings, to be approximately preserved in the current network.

### 3.3. Less-Forget Constraint

A model adapted to new data tends to forget what it has learned previously. Hence, one of the practical challenges for incremental learning is how to less forget the previous knowledge. To this end, we introduce a *less-forget constraint* through a new loss $L_{\mathrm{dis}}^{\mathrm{G}}$, which provides a stronger constraint on the previous knowledge compared to $L_{\mathrm{dis}}^{\mathrm{C}}$. Specifically, $L_{\mathrm{dis}}^{\mathrm{C}}$ mainly considers the local geometric structures, *i.e.* the angles between the normalized features and the old class embeddings. This constraint is

not able to prevent the embeddings and the features from being rotated entirely, as illustrated in Figure 4.

To enforce a stronger constraint on the previous knowledges, we propose to *fix* the old class embeddings and compute a novel distillation loss on the features as below:

$$L_{\text{dis}}^{\text{G}}(x) = 1 - \langle \bar{f}^*(x), \bar{f}(x) \rangle, \tag{6}$$

where $\bar{f}^*(x)$ and $\bar{f}(x)$ are respectively the normalized features extracted by the original model and those by the current one. $L_{\text{dis}}^{\text{G}}$ encourages the orientation of features extracted by current network to be similar to those by the original model. The loss is bounded ($L_{\text{dis}}^{\text{G}} \leq 2$). The rationale behind this design is that the spatial configuration of the class embeddings, to a certain extent, reflects the inherent relationships among classes. Hence, to preserve the previous knowledge, a natural idea is to keep this configuration. With the old class embeddings fixed, it is then reasonable to encourage the features to be similar as in $L_{\text{dis}}^{\text{G}}$.

In practice, as different numbers of new classes introduced in each phase (*e.g.* 10 classes *vs.* 100 classes), the degree of need to preserve the previous knowledge varies. In response to this, we propose to set the weight of the loss $L_{\text{dis}}^{\text{G}}$ (denoted as $\lambda$) adaptively as follows:

$$\lambda = \lambda_{\text{base}} \sqrt{|\mathcal{C}_{\text{n}}|/|\mathcal{C}_{\text{o}}|}, \tag{7}$$

where $|\mathcal{C}_{\text{o}}|$ and $|\mathcal{C}_{\text{n}}|$ are the number of old and new classes in each phase, $\lambda_{\text{base}}$ is a fixed constant for each dataset. In general, $\lambda$ increases when the ratio of the number of new classes to that of old classes increases.

Note that a recent work [19], which deals with the domain expansion that can be treated as one-phase incremental learning, also proposes to fix the last layer and mimic the features of the original model. However, our method differs from [19] in three aspects. (1) The distillation loss $L_{\text{dis}}^{\text{G}}$ only considers the orientation of the features but not the magnitudes (since the features are normalized in the loss), which gives more flexibility to the model to fit for new classes. (2) We introduce an adaptive coefficient to weight the distillation loss for more than one phase. (3) Our experiments show that the proposed method works well on long sequences of classes (*e.g.* 10 phases) and more realistic datasets (*e.g.* ImageNet), which have not been evaluated in [19].

### 3.4. Inter-Class Separation

Another practical challenge for multi-class incremental learning is how to form a unified classifier for all the classes, including both old and new ones, given that the data of new classes dominate the training set. In order to avoid the ambiguities between old and new classes, we introduce a margin ranking loss to ensure that they are well separated.

The reserved samples for old classes are fully exploited. Specifically, for each reserved sample $x$, we try to separate the ground-truth old class from all the new classes by a margin, using $x$ itself as an anchor. We consider the embedding of the ground-truth class as positive. To find the *hard* negatives, we propose an online mining method. We select those new classes that yield highest responses to $x$ as hard negative classes and use their embeddings as negatives for the corresponding anchor. Therefore, the proposed margin ranking loss is computed as:

$$L_{\text{mr}}(x) = \sum_{k=1}^{K} \max(m - \langle \bar{\theta}(x), \bar{f}(x) \rangle + \langle \bar{\theta}^k, \bar{f}(x) \rangle,\ 0), \tag{8}$$

where $m$ is the margin threshold, $\bar{\theta}(x)$ is the ground-truth class embedding of $x$, $\bar{\theta}^k$ is one of the top-$K$ new class embeddings chosen as hard negatives for $x$.

It is worth noting that, the positive and the negatives for each anchor are the class embeddings instead of samples. The proposed loss can be seamlessly incorporated in the training process without altering the data sampling process.

### 3.5. Integrated Objective

Our approach addresses the imbalance in multi-class incremental learning from multiple aspects. Combining the losses presented above, we reach a total loss comprised of three terms, given as:

$$L = \frac{1}{|\mathcal{N}|} \sum_{x \in \mathcal{N}} (L_{\text{ce}}(x) + \lambda L_{\text{dis}}^{\text{G}}(x)) + \frac{1}{|\mathcal{N}_{\text{o}}|} \sum_{x \in \mathcal{N}_{\text{o}}} L_{\text{mr}}(x), \tag{9}$$

where $\mathcal{N}$ is a training batch drawn from $\mathcal{X}$, $\mathcal{N}_{\text{o}} \subset \mathcal{N}$ are the reserved old samples contained in $\mathcal{N}$. $\lambda$ is a loss weight, which is set according to Eq (7).

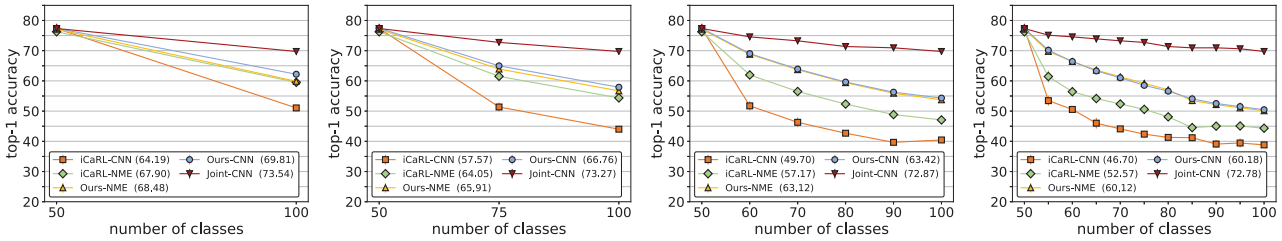
Besides, at the end of each training phase, we can further finetune the model with a balanced set of reserved samples taken from all observed classes. We find that the so-called *class balance finetune* can improve the performance moderately in pratice.

## 4. Experiment

### 4.1. Settings

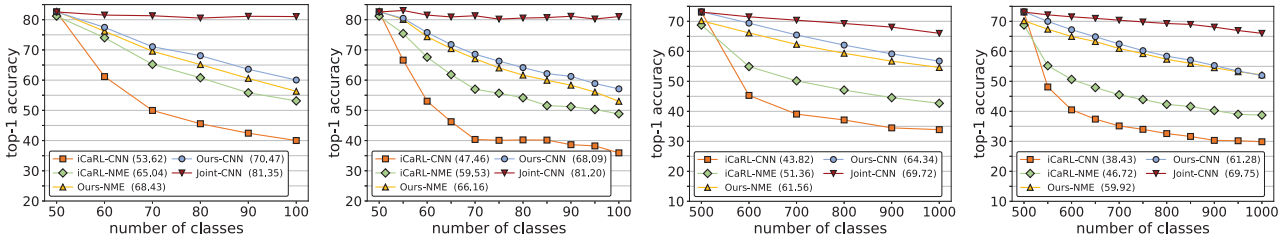
**Datasets.** Our experiments are conducted on two popular datasets for multi-class incremental learning, *i.e.* CIFAR100 [22], and ImageNet [7]. In a real-world application such as product categorization or face recognition, incremental learning usually starts from a model trained on a pre-collected dataset. To mimic this, we evaluate our algorithm starting from a model trained on half of classes for each dataset, and the rest classes come in different phases.

**Implementation Details.** All models are implemented with PyTorch and trained on TITAN-X GPUs. We adopt a 32-layer ResNet for CIFAR100 and a 18-layer ResNet for ImageNet. When adopting *cosine normalization* in the last




(a) CIFAR100 (1 phase) (b) CIFAR100 (2 phases) (c) CIFAR100 (5 phases) (d) CIFAR100 (10 phases)

Figure 5. The performance on CIFAR100. The average and standard deviations are obtained over three runs.




(a) ImageNet-Subset (5 phases) (b) ImageNet-Subset (10 phases) (c) ImageNet-Full (5 phases) (d) ImageNet-Full (10 phases)

Figure 6. The performance on ImageNet. Reported on ImageNet-Subset (100 classes) and ImageNet-Full (1000 classes).

layer, the ReLU in the penultimate layer is removed to allow the features to take both positive and negative values. For CIFAR100, the learning rate starts from 0.1 and is divided by 10 after 80 and 120 epochs (160 epochs in total). For ImageNet, the learning rates also starts from 0.1 and is divided by 10 every 30 epochs (90 epochs in total). Through the experiments, the networks are trained by SGD [23] with the batch size 128. The training images are randomly flipped and cropped as input, and no more data augmentation is used. For other hyper-parameters, $\lambda_{\text{base}}$ is set to 5 for CIFAR100 and 10 for ImageNet, $K$ is set to 2 and $m$ is set to 0.5 for all the experiments.

As for the strategy to reserve the samples for old classes, there are two popular ones. The first one stores a constant number of samples for each old class (*e.g.* $R_{\text{per}} = 20$), and thus the memory size grows with the number of classes. The second one considers a memory with fixed capacity (*e.g.* $R_{\text{total}} = 2000$ for CIFAR100 and $R_{\text{total}} = 20000$ for ImageNet). Since the capacity is independent of the number of classes, the more classes stored, the fewer samples reserved for each old class. In our experiments, we adopt the first strategy because it is usually more challenging (*e.g.* $R_{per} = 20$ *vs.* $R_{total} = 2000$ on CIFAR100) [2]. Besides, we used the method proposed in [29] based on *herd selection* [34] to select the samples to be reserved within each old class.

For the experiments on a given dataset, the classes are arranged in a fixed random order. Each method is then trained in a class-incremental way. After each incremental phase, the output model is evaluated on all the classes observed so far. Thus the evaluation result for each method is a curve of the classification accuracies after each phase. If a single number is preferable, we report the average of these accuracies, namely *average incremental accuracy* [29].

**Baselines.** iCaRL [29], as described in Section 3.1, is the representative method for *multi-class* incremental learning, which is adopted as the baseline here. More specifically, we respectively report its results of CNN predictions and *nearest-mean-of-exemplars* classification, denoted as *iCaRL-CNN* and *iCaRL-NME*.

For other methods, *Finetune* [11] and *Feature Extraction* [8] have proven to perform poorly for this setting [29, 3]. *LwF* [24] with a few additional reserved samples is equivalent to *iCaRL-CNN*, where the reserved samples have proven much helpful for incremental learning [29, 15]. Castro *et al*. [3] report better performance than iCaRL through the sophisticated data augmentation on the reserved old samples. However, according to the ablation study in [3], the performance is still inferior to iCaRL without the data augmentation. The recent works [35, 20] also report superior performance than iCaRL with the help of a generative model to produce the samples for old classes, which deal with the task in a different line with us and rely heavily on the quality of the generative model. The parameter-based methods such as EWC [21] and SI [37] have not evaluate

[2] We provide some results with the second strategy to reserve old samples in the supplementary material.

on ImageNet while MAS [1] and A-GEM [5] are evaluated in the *multi-task* setting.

To evaluate our model, we also respectively report the results achieved by the CNN predictions and *nearest-mean-of-exemplars* classification, denoted as *Ours-CNN* and *Ours-NME*. Besides, the results of *Joint Training* are provided as reference, which requires all previous data available in each phase [3].

### 4.2. Evaluation on CIFAR100

CIFAR100 is composed of 60000 images from 100 classes of size $32 \times 32$. Every class has 500 images for training and 100 images for evaluation. We start from a model trained on 50 classes and the remaining 50 classes come in 1, 2, 5 and 10 phases.

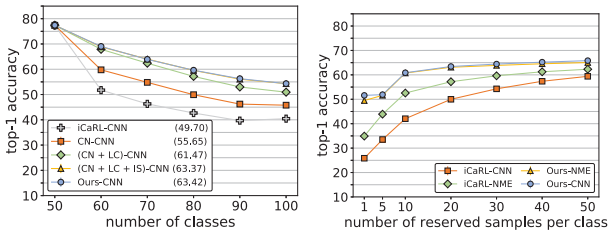
As shown in Figure 5, our method outperforms iCaRL by a large margin, either in the trend of classification accuracy curve or *average incremental accuracy*. Particularly, under the incremental setting of 10 phases (Figure 5(d)), the overall performance on the total 100 classes at the end of incremental learning is improved by more than 6% (*Ours-CNN vs. iCaRL-NME*). In our model, the CNN predictions performs (*i.e. Ours-CNN*) better or at least comparable to the *nearest-mean-of-exemplars* classification (*i.e. Ours-NME*), which is contrary to the observation in iCaRL [29]. Thus the CNN predictions can be directly adopted for predictions which indicates that the *imbalance* between old and new classes are well handled in our approach.

### 4.3. Evaluation on ImageNet

ImageNet is a large-scale dataset consisting of 1000 classes with more than 1000 images per class, which is a more challenging benchmark for incremental learning. In total, there are roughly 1.2 millon training images and 50k validation images. We report the performance on the validation set. Referring to [29, 3], we run two series of experiments on this dataset. In the first one, we conduct the experiments on a randomly selected subset of 100 classes, denoted as ImageNet-Subset. In the other one we evaluate our method on the whole 1000 classes denoted as ImageNet-Full. We start from a model trained on half of the total classes and divide the rest classes into 5 and 10 phases. The results are shown in Figure 6.

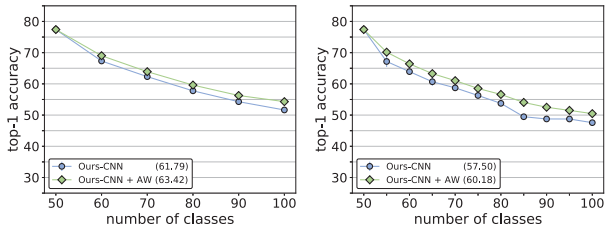
The observations on this dataset are consistent with those on CIFAR100. Our method performs significantly better than iCaRL under different settings. In our model, the results of CNN predictions is better or at least comparable to those of *nearest-mean-of-exemplars* classification. It is noteworthy that, under the incremental setting of 10 phases on ImageNet-Full, our method can reduce the overall classification error on the 1000 classes at the last phase by more than 13% (*Ours-CNN vs. iCaRL-NME* in Figure 6(d))

[3]We provide the results compared to more baselines in the supplementary material.




(a) CIFAR100 (5 phases) (b) CIFAR100 (5 phases)

Figure 7. (a) The effect of each component. (b) The effect of the number of reserved samples.



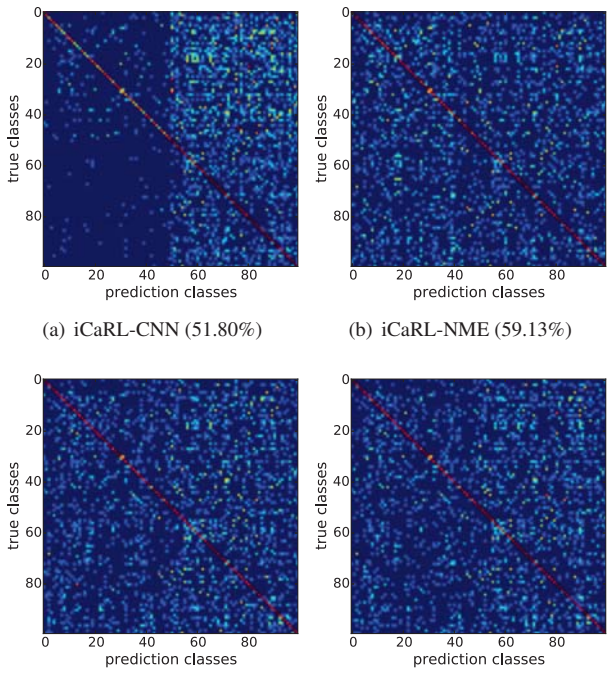

(a) CIFAR100 (5 phases) (b) CIFAR100 (10 phases)

Figure 8. The effect of adaptive loss weight (AW).

### 4.4. Ablation Study

**The effect of each component.** Our approach are mainly comprised of three components, *i.e. cosine normalization (CN)*, *less-forget constraint (LC)*, *inter-class separation (IS)*), When all the training is done, a *class balance finetune (CBF)* is further conducted on the reserved samples. Here we provide the results of some intermediate models to analyze the effect of each component: (a) *CN*: *cosine normalization* is adopted in the last layer and the distillation loss is updated as in Eq (5); (b) *CN + LC*: on the basis of *cosine normalization*, a stronger constraint is built to less forget the previous knowledge and the distillation loss is computed as in Eq (6); (c) *CN + LC + IS*: the proposed margin ranking loss in Eq (8) is further added to separate the old and new classes. For convenience, we only report the results of CNN predictions. From the results in Figure 7(a), we can observe that, each component has its contribution to the performance achieved by our final model, while *CBF* has a relatively small effect on this dataset since the adverse effects of the imbalance is mitigated by the former three components.

**The effect of the number of reserved samples.** To reserve a few samples have proven much helpful to maintain the performance for old classes [29, 15]. Figure 7(b) shows the comparison of our approach with iCaRL reserving d-




(a) iCaRL-CNN (51.80%) (b) iCaRL-NME (59.13%)

(c) Ours-NME (60.21%) (d) Ours-CNN (62.34%)

Figure 9. The comparison of confusion matrix (with entries transformed by $\log(1+x)$ for better visibility). CIFAR100 (1 phase) is adopted as the benchmark for convenience. Along with each method is the overall top-1 accuracy on the 100 classes.

ifferent number of samples per class. The more samples reserved, the better performance for both iCaRL and our approach. While in each case, the results of our approach are superior to those of iCaRL.

**The effect of adaptive loss weight.** In our approach we introduce an adaptive loss weight in Eq (7) for the distillation loss. Figure 8 shows the effect of adaptive loss weight compared to the baseline that uses the fixed constant $\lambda_{\text{base}}$[4] to weight the distillation loss. According to Figure 8, we can observe that, the adaptive loss weight for the distillation loss can help achieve better performance for long sequences of classes. Eq (7) is a heuristic strategy and we believe that there exists better choice to set the adaptive loss weight which will be explored in the future work.

**The comparison of confusion matrix.** Figure 9 shows the comparison of confusion matrix by iCaRL and our approach, which can provides further insight into the behaviors of both methods. *iCaRL-CNN* (Figure 9(a)) tends to classify the samples into new classes, while is caused by the severe imbalance between old and new classes. The adverse effects of the imbalance is mitigated in the last three methods, while *Ours-CNN* achieves the best overall performance. The confusion matrix of *Ours-CNN* suggests more balanced predictions over all classes, both in terms of diagonal entries (*i.e.* correction predictions) as well as off-diagonal entries (*i.e.* mistakes), which indicates that the class imbalance is well handled in our approach.

[4] $\lambda_{\text{base}}$ is optimized in the case of 1 phase where the number of old and new classes are the same.

## 5. Conclusion

This work develops a novel framework to learn a unified classifier under the *multi-class* incremental setting. Our study reveals that the imbalance between old and new classes is an crucial cause for the challenges in this task, which is handled from different aspects in our approach, including *cosine normalization*, *less-forget constraint*, and *inter-class separation*. The combination of these components rebalances the training process which can thus more effectively preserve the previous knowledge and reduce the ambiguities between old and new classes. The extensive experiments on CIFAR100 and ImageNet demonstrate that our approach outperforms iCaRL by a large margin, and brings consistent improvements under different settings.

## Acknowledgment

This work is partially supported by the NSFC under Grant 61673362 & 61836008, Youth Innovation Promotion Association CAS, and the Fundamental Research Funds for the Central Universities. This work is partially supported by the Collaborative Research grant from SenseTime Group (CUHK Agreement No. TS1610626 & No. TS1712093), and the General Research Fund (GRF) of Hong Kong (No. 14236516 & No. 14203518).

## References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, 2018.

[2] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *CVPR*, 2017.

[3] Francisco M Castro, Manuel Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *ECCV*, 2018.

[4] Gert Cauwenberghs and Tomaso Poggio. Incremental and decremental support vector machine learning. In *NIPS*, 2001.

[5] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *ICLR*, 2019.

[6] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority oversampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[8] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014.

[9] Qi Dong, Shaogang Gong, and Xiatian Zhu. Class rectification hard mining for imbalanced deep learning. In *ICCV*, 2017.

[10] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, 2018.

[11] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[12] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *International Joint Conference on Neural Networks*, 2008.

[13] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge & Data Engineering*, (9):1263–1284, 2008.

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[15] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Lifelong learning via progressive distillation and retrospection. In *ECCV*, 2018.

[16] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *CVPR*, 2016.

[17] Chen Huang, Chen Change Loy, and Xiaoou Tang. Discriminative sparse neighbor approximation for imbalanced learning. *IEEE transactions on neural networks and learning systems*, 29(5):1503–1513, 2018.

[18] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

[19] Heechul Jung, Jeongwoo Ju, Minju Jung, and Junmo Kim. Less-forgetful learning for domain expansion in deep neural networks. In *AAAI*, 2018.

[20] Ronald Kemker and Christopher Kanan. Fearnet: Brain-inspired model for incremental learning. In *ICLR*, 2018.

[21] 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.

[22] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[24] Zhizhong Li and Derek Hoiem. Learning without forgetting. In *ECCV*, 2016.

[25] Chunjie Luo, Jianfeng Zhan, Xiaohe Xue, Lei Wang, Rui Ren, and Qiang Yang. Cosine normalization: Using cosine similarity instead of dot product in neural networks. In *International Conference on Artificial Neural Networks*, 2018.

[26] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *arXiv preprint arXiv:1802.07569*, 2018.

[27] Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *CVPR*, 2018.

[28] Amal Rannen Ep Triki, Rahaf Aljundi, Matthew Blaschko, and Tinne Tuytelaars. Encoder based lifelong learning. In *ICCV*, 2017.

[29] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017.

[30] Stefan Ruping. Incremental learning with support vector machines. In *ICDM*, 2001.

[31] 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.

[32] Kai Ming Ting. A comparative study of cost-sensitive boosting algorithms. In *ICML*, 2000.

[33] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NIPS*, 2016.

[34] Max Welling. Herding dynamical weights to learn. In *ICML*, 2009.

[35] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, Zhengyou Zhang, and Yun Fu. Incremental classifier learning with generative adversarial networks. *arXiv preprint arXiv:1802.00853*, 2018.

[36] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *ICLR*, 2018.

[37] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, 2017.

[38] Zhi-Hua Zhou and Xu-Ying Liu. On multi-class cost-sensitive learning. In *AAAI*, 2006.