Defying Imbalanced Forgetting in Class Incremental Learning

Shixiong Xu[1,2], **Gaofeng Meng**[1,2,3*], **Xing Nie**[1,2], **Bolin Ni**[1,2], **Bin Fan**[4], **Shiming Xiang**[1,2]

[1]State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]Centre for Artificial Intelligence and Robotics, HK Institute of Science & Innovation, Chinese Academy of Sciences
[4]School of Intelligence Science and Technology, University of Science and Technology, Beijing
{xushixiong2020, niexing2019, nibolin2019}@ia.ac.cn, bin.fan@ieee.org, {gfmeng, smxiang}@nlpr.ia.ac.cn

Abstract

We observe a high level of imbalance in the accuracy of different classes in the same old task for the first time. This intriguing phenomenon, discovered in replay-based Class Incremental Learning (CIL), highlights the imbalanced forgetting of learned classes, as their accuracy is similar before the occurrence of catastrophic forgetting. This discovery remains previously unidentified due to the reliance on average incremental accuracy as the measurement for CIL, which assumes that the accuracy of classes within the same task is similar. However, this assumption is invalid in the face of catastrophic forgetting. Further empirical studies indicate that this imbalanced forgetting is caused by conflicts in representation between semantically similar old and new classes. These conflicts are rooted in the data imbalance present in replay-based CIL methods. Building on these insights, we propose CLass-Aware Disentanglement (CLAD) to predict the old classes that are more likely to be forgotten and enhance their accuracy. Importantly, CLAD can be seamlessly integrated into existing CIL methods. Extensive experiments demonstrate that CLAD consistently improves current replay-based methods, resulting in performance gains of up to 2.56%.

Introduction

In typical image recognition tasks, the data is assumed to follow the independently and identically distributed (i.i.d.) assumption. A good classification model is expected to have similar and high accuracy across different classes. But in the real world, the data is non-stationary. To address this issue, Class Incremental Learning allows the model to continually learn new classes without forgetting the previously learned ones (Van de Ven and Tolias 2019). However, when sequential fine-tuning is performed on new classes without the presence of old data, there is a dramatic drop in accuracy for the learned tasks, known as catastrophic forgetting (McCloskey and Cohen 1989).

As the accuracy between the old and new classes is highly imbalanced in CIL, numerous approaches have been proposed. Among them, exemplar replay has been proven to be a simple yet effective strategy (Robins 1995; Rebuffi et al. 2017; Riemer et al. 2018; Buzzega et al. 2020). In exemplar replay, a subset of each class is selected and stored in a buffer. During the training of subsequent tasks, these exemplars are reused in various ways to help preserve the learned knowledge, such as joint training (Robins 1995; Riemer et al. 2018), knowledge distillation (Hou et al. 2019; Rebuffi et al. 2017), gradient projection (Lopez-Paz and Ranzato 2017; Saha, Garg, and Roy 2021; Deng et al. 2021), and bias correction (Hou et al. 2019; Wu et al. 2019; Zhao et al. 2020; Prabhu, Torr, and Dokania 2020). It is important to note that all these efforts primarily focus on tackling the accuracy imbalance between old and new classes. However, we argue that these efforts alone are insufficient to meet the expectations of a classification model.

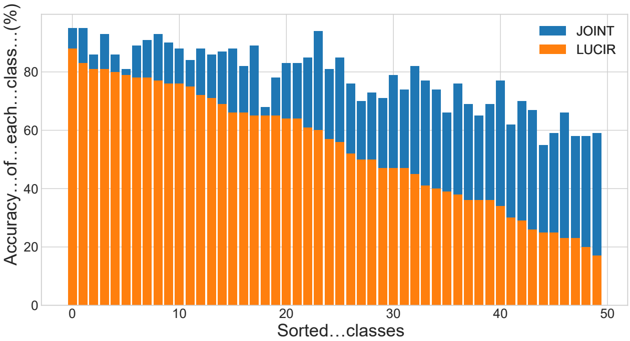

Figure 1: Demonstration of imbalanced forgetting. Visualization of the accuracy of each class in the first task obtained by joint training and LUCIR (Hou et al. 2019). The class indexes are sorted according to the result from LUCIR for better visualization.More illustrations of the above phenomenon with other methods can be found in the supplementary material.

For the first time, we observed that the accuracy between classes of the same old task is also highly imbalanced. It is evident that **imbalanced forgetting** occurs during the learning of a new task, as their accuracy is similar just after learning (as shown in Fig. 1 JOINT). This phenomenon remains undiscovered because the average incremental accuracy used for measuring CIL approaches assumes that the accuracy of the classes within the same task is uniform. Fig. 1 provides an example where the model is trained using a CIL setting that splits CIFAR-100 (Krizhevsky et al. 2009) into six tasks, with 50 classes in the first task and

[*]Corresponding author.

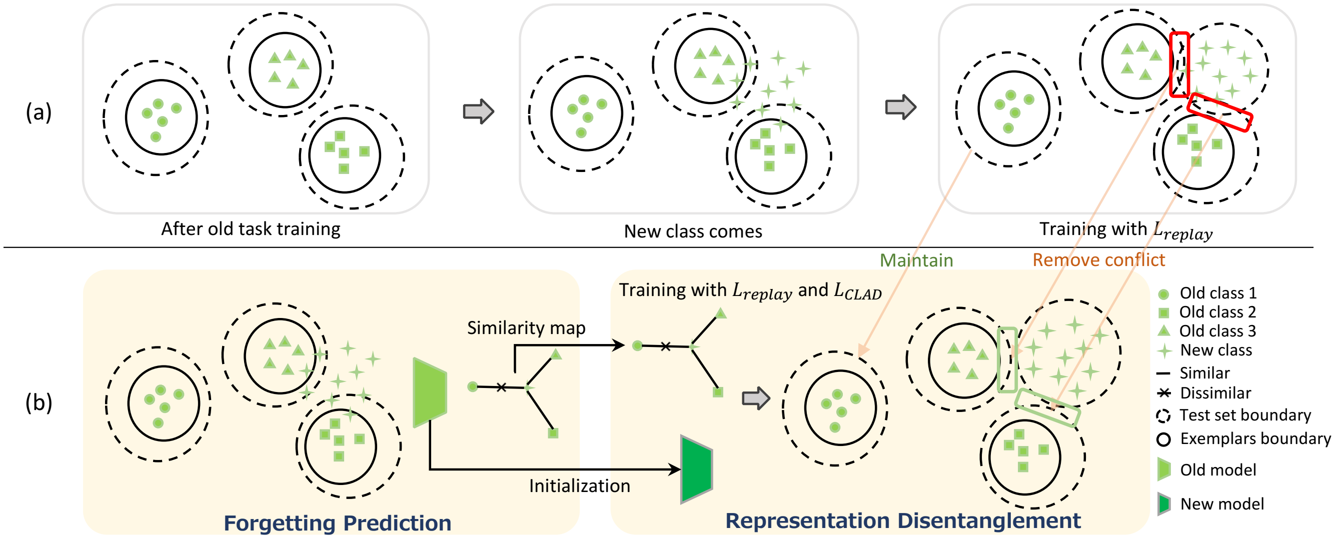

Figure 2: An overview of (a) existing replay-based methods and (b) our proposed CLAD. In existing replay-based methods, different old classes 1,2, and 3 have different accuracy because of the different similarities with the new class. The limited exemplars are not sufficient to preserve the boundary of the test set (low accuracy of classes 2 and 3 in (a)). Our proposed CLAD consists of two parts: Forgetting Prediction (FP) and Representation Disentanglement (RD). FP aims to find the classes that might be forgotten during the learning of new classes (classes 2 and 3). Based on the similarity information from FP, RD encourages the representation of new classes to stay away from similar old ones.

10 classes per subsequent task. The accuracy of each class in the first task is reported. From the perspective of representation learning, the low accuracy of some classes in joint training results from their semantic similarity. In the context of CIL, the forgetting of old classes occurs due to the introduction of similar new classes (Ramasesh, Dyer, and Raghu 2020). This explains why imbalanced forgetting occurs. For example, let's consider there are two classes, {`male, dog`} in the first task, and the model learns a new class, `female`. The `male` class tends to forget more easily due to representation interference.

The above observations provide insight into the potential for improvement in average accuracy lying in the classes that are easy to forget. From a methodological perspective, the key is to minimize interference with the representations of those old vulnerable classes when new tasks arise. This raises two questions: 1) how to identify the vulnerable old classes and 2) how to mitigate conflicts between the representations of old and new classes. Regarding the first question, a positive relationship between inter-class similarity and class-level forgetting is established in conventional CIL settings. Therefore, it is possible to predict the forgetting level of old classes based on their similarity with the new classes. Building on this, a novel framework called *CLass-Aware Disentanglement* (CLAD) is proposed to address the second question. As illustrated in Fig. 2 (b), CLAD consists of two phases: Forgetting Prediction (FP) and Representation Disentanglement (RD). In the FP phase, a subset of the old classes is identified as vulnerable ones, where conflicts are likely to occur with new classes. Empirical demonstrations and statistical analyses indicate a strong relationship between the conflict classes identified by FP and the degree of forgetting (see Fig. 3). During training for the new task, RD is introduced to constrain the similarity between the representations of samples in the new classes and the exemplars of their corresponding conflict classes. CLAD is formulated as a regularization term, which can be incorporated as a plugin for existing replay-based methods.

Extensive experiments on CIFAR-100 (Krizhevsky et al. 2009) and ImageNet (Deng et al. 2009) indicate that CLAD provides a consistent and impressive performance improvement over existing methods. Besides, comprehensive ablation studies are performed to show how the components in CLAD, including the conflict classes selection, regularization coefficient, and buffer size, influence its performance.

Our contributions are summarized as follows:

- To the best of our knowledge, we are the first to reveal the imbalanced forgetting between the learned classes.
- Experiments and statistical analysis are conducted to demonstrate that imbalanced forgetting results from varying semantic similarity between inter-task classes.
- CLass-Aware Disentanglement (CLAD) is proposed to improve the accuracy of vulnerable old classes, which can be used as a plugin for replay-based CIL methods.
- Extensive experiments on several challenging benchmarks demonstrate that CLAD can provide consistent improvements over existing methods.

Related Work

Typically, since only the data of the current task is available in each training phase, the main challenge of CIL is performance deterioration for old classes, *i.e.* catastrophic forgetting (McCloskey and Cohen 1989). An intuitive way is using a buffer to save some exemplars of each old class, and training the exemplars along with new data to mimic the i.i.d. joint training, named Experience Replay (Robins 1995; Riemer et al. 2018). However, this strategy leads to a severe imbalance between current and old classes, and the bias towards new classes still exists (Hou et al. 2019; Wu et al. 2019; Zhao et al. 2020; Prabhu, Torr, and Dokania 2020).

Many approaches were proposed to further utilize the exemplars in recent years (Li and Hoiem 2017; Hou et al. 2019; Rebuffi et al. 2017; Douillard et al. 2020; Lopez-Paz and Ranzato 2017; Chaudhry et al. 2018; Farajtabar et al. 2020; Saha, Garg, and Roy 2021; Wu et al. 2019; Prabhu, Torr, and Dokania 2020), which can be split into three categories, knowledge distillation (Li and Hoiem 2017; Hou et al. 2019; Rebuffi et al. 2017; Douillard et al. 2020), gradient projection (Lopez-Paz and Ranzato 2017; Chaudhry et al. 2018; Farajtabar et al. 2020; Saha, Garg, and Roy 2021; Deng et al. 2021), and bias correction (Hou et al. 2019; Wu et al. 2019; Zhao et al. 2020; Prabhu, Torr, and Dokania 2020), respectively. Knowledge distillation is a training strategy first proposed by (Hinton et al. 2015) to transfer the knowledge from the teacher model to the student model. LwF (Li and Hoiem 2017) uses this technology to preserve the knowledge of the old model (teacher model) during the new task training for the first time. Subsequent methods (Hou et al. 2019; Rebuffi et al. 2017; Douillard et al. 2020) further involve the replay buffer and multiple distillation loss in CIL. Gradient projection methods try to keep the gradient of new tasks from interfering with old ones through projection, represented by GEM (Lopez-Paz and Ranzato 2017), A-GEM (Chaudhry et al. 2018), GPM (Saha, Garg, and Roy 2021), FSDGPM (Deng et al. 2021), and OGD (Farajtabar et al. 2020). Inspired by the similarity between class imbalanced learning and CIL with exemplar replay (He, Wang, and Chen 2021), BiC (Wu et al. 2019) and WA (Zhao et al. 2020) demonstrate that the bias occurs in the weights of classifier, and attempt to correct the bias by post-processing the weights. However, bias also exists in the backbone of the network. GDumb (Prabhu, Torr, and Dokania 2020) tackles this imbalance by constructing balanced data during training.

Besides the end-to-end methods (Li and Hoiem 2017; Hou et al. 2019; Douillard et al. 2020; Wu et al. 2019), several plugin methods emerging recently for CIL are based on the observation of the i.i.d. training process or results (Shi et al. 2022; Ashok, Joseph, and Balasubramanian 2022; Liu, Schiele, and Sun 2021). CwD (Shi et al. 2022) enforces the data representations to be more uniformly scattered at the first task, which mimics the representation extracted by the model trained with all classes (oracle model). CSCCT (Ashok, Joseph, and Balasubramanian 2022) proposes two regularization terms to cluster and distillate the class features, and encourage new classes to be situated optimally in the feature space. AANet (Liu, Schiele, and Sun 2021) proposes to use a new branch for stable knowledge learning, which is effective but needs more memory.

Discussion. Most of the methods treat all the old classes equally and attempt to provide end-to-end methods to overcome the catastrophic forgetting problem. Differently, we try to mitigate the class-level representation interference in a class-aware way, which is inspired by our observation that the forgetting of different old classes is severely imbalanced. A similar effort has been made by LUCIR (Hou et al. 2019), which adopts a margin ranking loss to encourage a large margin between the logits of old and new classes. However, it calculates the similarity for each sample, which will lead to inconsistent class-level similarity prediction. And it is not conducive to clustering together the representations of the same class. Our proposed Class-Aware Disentanglement (CLAD) measures the representation interference with cosine similarity and calculates the class similarity at the class level, which is more robust for CIL.

Methodology

Preliminary

Formally, the sequentially trained model is denoted by $f(\cdot) = g(\phi(\cdot))$, which consists of a feature extractor $\phi(\cdot)$ followed by a classification layer $g(\cdot)$. $f(\cdot)$ is trained on sequential tasks $\mathcal{T} = [\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_T]$ with non-overlapping classes. At the t-th training task, the model after training on the $(t-1)$-th task is denoted by f_{t-1} and will be incrementally trained on the new dataset $\mathcal{D} = \mathcal{D}_t \cup \mathcal{B}$. The buffer $\mathcal{B} \subset [\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_{t-1}]$ keeps only several exemplars of each old class. Under this setting, the standard cross-entropy loss of replay-based CIL methods is formulated as:

$$\mathcal{L}_{ce} = -\frac{1}{\mid \mathcal{D} \mid} \sum_{k=1}^{|\mathcal{D}|} y_k \log(\sigma(f(x_k))), \tag{1}$$

where (x_k, y_k) is an image and its label in $\mathcal{D}$.

However, as the training data is highly imbalanced, L_{ce} is not a good approximation of standard joint training classification loss at the t-th task. Therefore, various additional constraints are proposed to help CIL methods better approximate the ideal loss during incremental training, such as knowledge distillation (Li and Hoiem 2017; Rebuffi et al. 2017; Hou et al. 2019), re-sampling (Prabhu, Torr, and Dokania 2020; Wu et al. 2019; Zhao et al. 2020), and gradient projection (Saha, Garg, and Roy 2021; Chaudhry et al. 2018; Deng et al. 2021). In general, denote these additional constraints by L_{ad}, the complete loss function $\mathcal{L}_{replay}$ of replay-based methods is formulated as:

$$\mathcal{L}_{replay} = \mathcal{L}_{ce} + \lambda \mathcal{L}_{ad}, \tag{2}$$

where λ stands for the adjustable weight of $\mathcal{L}_{ad}$.

What Causes Imbalanced Forgetting?

In this section, we try to reveal the relationship between inter-class similarity and class-level forgetting. Intuitively, semantically similar classes are more likely to be misclassified from each other in standard model training with i.i.d. data. In the CIL setting with replay buffer, we can further

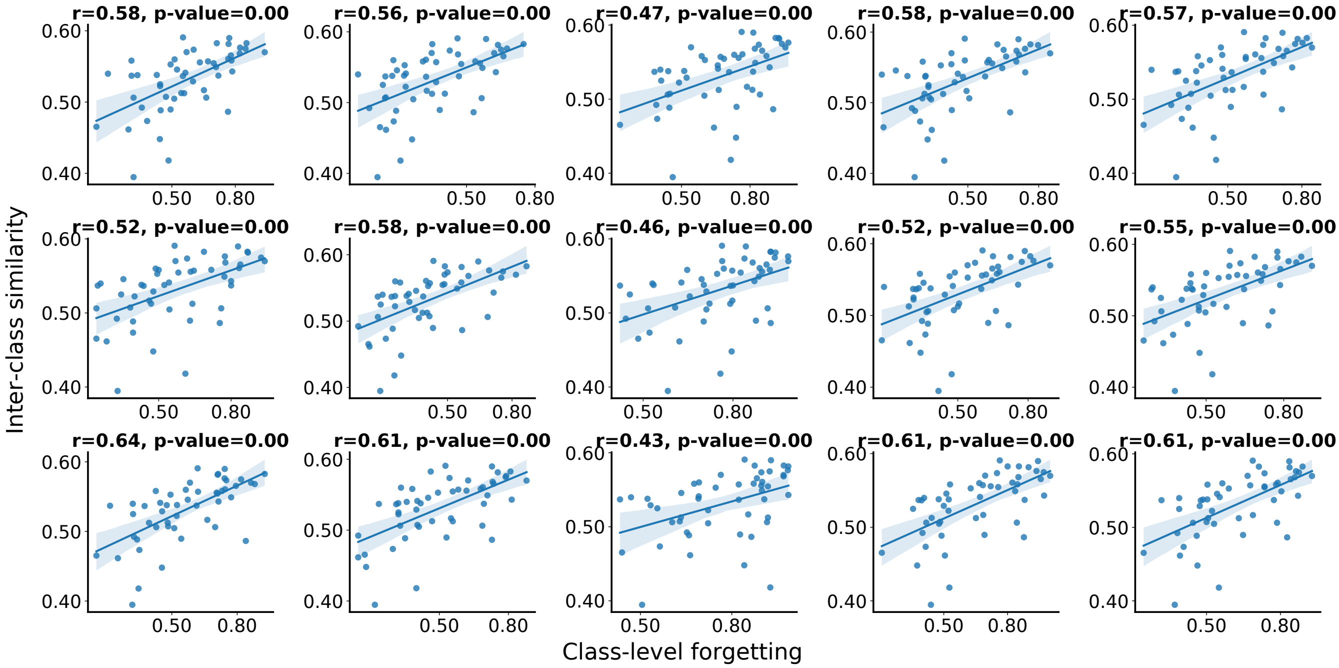

Figure 3: Illustration of the relative class forgetting and average similarity with latter classes for each old one. There is a positive correlation between the maximum similarity forgetting in different settings and methods. The first row gives the experiments that begin with 50 classes and 10 classes for each latter task, and the number of classes in the latter tasks in the second and third rows is 5 and 2, respectively.

infer that *given a class the in old task, if there are classes very similar to it in the latter tasks, it is more likely to suffer performance degradation and vice versa*.

To verify this hypothesis, three task sequences with different lengths are constructed using CIFAR-100 (Krizhevsky et al. 2009). All three sequences begin with a base task containing 50 classes, and the number of classes in the subsequent tasks is 10, 5, and 2, respectively. Then 12 models are trained on the above three tasks with four representative CIL methods: naive replay with loss function defined in Eq. 1, LUCIR (Hou et al. 2019), BiC (Wu et al. 2019), and iCaRL (Rebuffi et al. 2017). The hyperparameters of all the experiments are consistent with the original papers, and the number of exemplars for each class is 20 as the common practice in (Rebuffi et al. 2017; Hou et al. 2019; Liu, Schiele, and Sun 2021; Douillard et al. 2020). The accuracy of each class after learning each task is recorded. Formally, we denote the accuracy of class i after training on the *base* task by A^i_{base} and that after training on the *all* tasks is denoted as A^i_{all}. As A^i_{base} is different for each class i, the class-level forgetting δ_i is defined by normalized accuracy drop:

$$\delta_i = \frac{A^i_{base} - A^i_{all}}{A^i_{base}}. \tag{3}$$

To explore the relationship between the class-level forgetting δ_i and inter-class similarity, a reasonable approach is needed to measure the similarities between class i and the latter 50 classes. A straightforward way is to train an **oracle** model on all the classes or even larger datasets, then use the features extracted by it to calculate inter-class cosine similarity (Shi et al. 2022). Although this pipeline is standard and widely used in other fields like image retrieval (Chen et al. 2021) and multi-modal learning (Baltruaitis, Ahuja, and Morency 2017), two flaws limit its use in the CIL:

- In the CIL setting, an oracle model is unavailable even after seeing all the classes.
- The representation space of the oracle model obtained by joint training is highly different from the changing one in incremental learning.

To address the above limitations, we opt to use the model trained on the first 50 classes (denoted by f_1) instead of the oracle model for similarity calculation. The model f_1 is available after training of the first task, and the feature space aligns well with all the learned 50 classes naturally. Furthermore, the logits of the latter 50 classes obtained by f_1 are used as the similarity between old and new classes, which is essentially equivalent to the cosine similarity without normalization but much simpler in terms of calculation and implementation. Formally, the inter-class similarity level of each old class is defined as $S = Mean(f_1(C))$, where C indicates a given new class. The inter-class similarity for old class i and class C is denoted by $S_i = S[i]$. The functional equivalence between cosine similarity and logits similarity is shown in Tab. 3.

With the above preparation, the relationship between inter-class similarity S_i and class-level forgetting δ_i could be established. As shown in Fig. 3, there is an obvious positive correlation between the two variables under different settings and baselines. **Concretely, the Pearson correlation of them reached 0.6 with high confidence (p-value=0.00)**. This reveals that in the replay-based method, the representation of the class in the previous task might clash with the class representation in the new task similar to it, resulting in more pronounced forgetting. We would like to emphasize again that the purpose of this subsection is to *establish* the relationship to find a method or a metric to predict where forgetting will happen.

Class-Aware Disentanglement

Motivated by the above observations and (Ramasesh, Dyer, and Raghu 2020), given a new class from the current task, encouraging its representation to keep distance from the old similar classes is beneficial to alleviate the forgetting of the corresponding old classes. To achieve this, two issues need to be addressed. First, how to predict classes that are most likely to be forgotten in the task sequence. Second, how to effectively separate the representations of the corresponding classes during the training process. In the subsequent content, the above two questions will be answered in turn. And the overview of the proposed CLAD is shown in Fig. 2 (b).

Forgetting Prediction (FP). As stated above, we could predict which old classes in the first task are more likely to be forgotten with the model f_1. Extending this idea to the whole task sequence, the learned classes that are vulnerable to forgetting after the $(t-1)$-th task can be predicted by model f_{t-1}. Formally, for a new class C in task t, the similarity between it and all the old classes is formulated as:

$$S(C) = \frac{1}{\mid C \mid} \sum_{k=1}^{|C|} f_{t-1}(x_k), \tag{4}$$

where $S(C)$ is the mean logit vector of class C obtained by the model f_{t-1}. $|C|$ indicates the number of samples in class C and x_k is the k-th sample in class C. By sorting the $S(C)$, we can predict which old classes are most likely to forget when learning new class C.

Representation Disentanglement (RD). Equipped with FP, for a new class, we can locate which old classes are most likely to be forgotten. Naturally, forgetting can be mitigated by disentangling the learned representation of the new class and that of the corresponding old classes.

Denoting the number of learned classes by N, a fixed proportion of the old classes are selected for new class C as the conflict classes. This proportion $\mathcal{P}$ is defined as the conflict proportion. Given a sample x from class C, the conflict classes are obtained by selecting the indexes of old classes that have the Top-$(\mathcal{P} * N)$ largest values in $S(C)$.

Now the conflict between the new classes and old classes can be mitigated by disentangling their representations. The new task is trained on joint data $\mathcal{D} = \mathcal{D}_t \cup \mathcal{B}$ with replay-based methods. In each iteration, the data batch contains both new and old classes. Based on this, a novel class-aware disentanglement regularization is proposed. Considering the representation distribution of the old classes is shifting during the new task training, the representation conflict is disentangled in both online and offline ways. In online disentanglement, new class representations are encouraged to be separated from the online representations of the conflict classes in the same batch. In offline disentanglement, the representations of class C are encouraged to keep their distance from the old representations of all conflict classes in the buffer. Given a sample x and its conflict samples X_o in the same batch and X_b in the buffer, the CLAD loss is formulated as:

$$\mathcal{L}_{on}(x) = \frac{1}{\mid X_b \mid} \sum_{x_b \in X_b} (1 + cos(\phi(x), \phi(x_b))), \tag{5}$$

$$\mathcal{L}_{off}(x) = \frac{1}{\mid X_o \mid} \sum_{x_o \in X_o} (1 + cos(\phi(x), \phi_{t-1}(x_o))), \tag{6}$$

where $cos(\cdot, \cdot)$ indicates the cosine similarity. Thus the objective of CLAD is:

$$\mathcal{L}_{CLAD} = \frac{1}{\mid C \mid} \sum_{x \in C} (\mathcal{L}_{\text{on}}(x) + \mathcal{L}_{\texttt{off}}(x)). \tag{7}$$

Accordingly, the overall loss function is written as:

$$\mathcal{L} = \mathcal{L}_{replay} + \eta \mathcal{L}_{CLAD}, \tag{8}$$

where η is the coefficient of CLAD loss.

Experiments

Experimental Setup

Datasets and Protocols. Three commonly used benchmarks (Hou et al. 2019) are selected to evaluate the proposed method. CIFAR-100 (Krizhevsky et al. 2009) consists of 600,000 images from 100 classes, and the image size is 32×32. ImageNet (Deng et al. 2009) contains about 1.2 million 224×224 RGB images from 1000 classes. ImageNet-100 (Rebuffi et al. 2017) is a subset of ImageNet (Deng et al. 2009), which is sampled as (Hou et al. 2019; Liu, Schiele, and Sun 2021). To be consistent with the protocols of the previous work (Hou et al. 2019; Rebuffi et al. 2017; Liu, Schiele, and Sun 2021; Liu et al. 2020; Hu et al. 2021), all the classes of each dataset are shuffled with seed 1993 before splitting them into tasks. For CIFAR-100 and ImageNet-100, half classes are selected for the first task to mimic the pre-collected dataset in real-world (Hou et al. 2019), then there are $S = 10/5/2$ classes for each latter task. For ImageNet, 100 classes are selected for the first task, then the model learns 100 or 50 classes per task incrementally.

Metrics. The average incremental accuracy (A_t) (Douillard et al. 2020; Hu et al. 2021; Hou et al. 2019) is used to evaluate the performance of the baselines and our results. Formally, denote the test accuracy of the model after the training of the i-th task as A_i, then the average incremental accuracy after the t-th task is defined as $A_t = \frac{1}{t} \sum_{i=1}^{t} A_i$.

Implementation details. Following the previous studies (Shi et al. 2022; Yan, Xie, and He 2021), we adopt ResNet-18 (He et al. 2016) for all the experiments bellow. Notably, for CIFAR-100 (Krizhevsky et al. 2009) the kernel size of the first convolution layer is set to 3×3, and the following maxpooling layer is removed for higher feature resolution (Shi et al. 2022). And SGD is used as the optimizer. The learning rate is set to 0.1, the batch size is

Method	CIFAR-100 (B=50)			ImageNet-100 (B=50)			ImageNet (B=100)	
	S=10	5	2	10	5	2	100	50
LwF (Li and Hoiem 2017)	54.01	48.40	45.49	54.22	48.95	43.29	41.42	28.31
iCARL (Rebuffi et al. 2017)	67.16	60.54	54.50	72.03	68.23	59.54	49.88	42.52
BiC (Wu et al. 2019)	63.11	56.27	48.83	70.09	64.88	57.82	52.46	47.30
LUCIR (Hou et al. 2019)	66.16	60.43	52.22	70.40	67.19	62.86	52.47	47.55
+ CLAD	67.57$_{+1.41}$	62.15$_{+1.72}$	53.51$_{+1.29}$	73.05$_{+2.65}$	68.34$_{+1.15}$	64.24$_{+1.38}$	53.36$_{+0.89}$	48.79$_{+1.24}$
PODNet (Douillard et al. 2020)	68.56	65.57	62.95	75.90	72.41	65.28	56.86	53.68
+ CLAD	69.07$_{+0.51}$	65.96$_{+0.39}$	63.29$_{+0.34}$	76.02$_{+0.12}$	73.10$_{+0.69}$	65.45$_{+0.17}$	57.36$_{+0.50}$	55.38$_{+1.70}$
CwD (Shi et al. 2022)	66.81	61.86	56.41	71.43	68.92	65.06	52.56	47.88
+ CLAD	67.76$_{+0.95}$	63.67$_{+1.81}$	57.79$_{+1.38}$	72.33$_{+0.90}$	70.01$_{+1.09}$	65.92$_{+0.86}$	53.64$_{+1.08}$	49.07$_{+1.19}$

Table 1: The improvement achieved by adding CLAD to the SOTAs (Hou et al. 2019; Shi et al. 2022; Douillard et al. 2020) and the comparison with three baselines (Li and Hoiem 2017; Wu et al. 2019; Rebuffi et al. 2017). B and S denote the number of classes in the first task and the subsequent tasks. All the results are reproduced with the source code from (Shi et al. 2022).

set to 128, the momentum is set to 0.9, and the weight decay is 5e-4. For CIFAR-100, all the methods are trained for 160 epochs for each task, and the learning rate is multiplied by 0.1 at the 80-th and 120-th epoch. For ImageNet and ImageNet100, the models are trained for 90 epochs for each task, and the learning rate is multiplied by 0.1 as the 30-th and 60-th epoch. Since we focus on the replay-based methods, the *Herding* strategy is used to select the exemplars for replay after training each task (Rebuffi et al. 2017), and the number of exemplars per class is 20, which is consistent with (Hou et al. 2019; Shi et al. 2022). The conflict proportion is set to 0.1 for all experiments empirically. The CLAD coefficient is set to 4 for LUCIR and CwD, while the value of it is 2 for PODNet. How to determine these values is detailed in the supplementary material.

Improvements over Baselines

We add our proposed CLAD to three strong CIL baselines: LUCIR (Hou et al. 2019), CwD (Shi et al. 2022) and PODNet (Douillard et al. 2020). Tab. 1 shows the results on CIFAR-100, ImageNet-100, and ImageNet. Our method provides consistent improvement of average incremental accuracy by around 0.5% and 2.5% on various datasets and settings, *e.g.*, on CIFAR-100, LUCIR with CLAD gains up to 1.72% on accuracy while CwD with CLAD improves the baseline by 1.81% at most. On ImageNet-100 when $C = 10$, CLAD makes the LUCIR (Hou et al. 2019) even outperform the stronger baseline CwD (Shi et al. 2022). CLAD also provides similar performance improvements on larger datasets ImageNet, indicating that our method adapts well to larger datasets. Besides, one may notice that the improvements on PODNet (Douillard et al. 2020) are limited compared with those on LUCIR and CwD. This phenomenon can be explained by the special design of PODNet, which conducts feature distillation even for middle layers to help preserve knowledge. However, our CLAD loss only disentangles the features in the final layer, which is consistent with the distillation loss in LUCIR and CwD. Although distillation does a good job of mitigating forgetting, the distillation with new samples will enhance the representation conflict in the middle layer in PODNet (Chen et al. 2023).

Ablation Study

In this subsection, extensive ablation studies are conducted to analyze the key components of CLAD. If not specified, all the experiments are based on LUCIR (Hou et al. 2019) under the protocol that split CIFAR-100 (Krizhevsky et al. 2009) into six tasks with 50 classes for the first task and 10 classes for the rest. The results are averaged over three runs.

Effectiveness of conflict prediction. Although we are committed to mitigating representation conflict between similar old and new classes, it is doubtful that mitigating conflicts between arbitrary old and new classes also help CIL. To dispel this concern, apart from selecting the Top-$\mathcal{P}N$ largest values in $S(C)$ as old classes, we select the Top-$\mathcal{P}N$ smallest values in $S(C)$ and randomly select $\mathcal{P}N$ old classes for comparison, named as *Smallest* and *Random*. As shown in Fig. 4 (a), the latter two strategies are harmful no matter what proportion of the old classes is chosen. Because the CLAD loss constrains the feature distribution of the new class to some extent. But the compromise of new classes will not benefit the performance of the old class when there is no representation conflict between them, which happens in *Smallest* and *Random* strategies.

Proportion of conflict classes. In this part, we further investigate the impact of different proportions of chosen conflict classes for each new class. In Fig. 4 (a), it is shown that the improvement gained by CLAD is relatively small when the proportion is extremely large or small, while CLAD achieves the greatest performance gain with a proportion set to 0.1. This observation is intuitive because too few conflict classes are not sufficient for mitigation and there can be prediction errors. Too many conflicting classes will hurt the performance of the new class more because some old classes do not conflict with the new class.

Components in conflict mitigation. The proposed CLAD loss has two components as stated above, now we ablate each component of CLAD in Fig. 4 (c). Each of the components can improve the average incremental accuracy of the baseline, and the combination of both can further help improve the performance.

Improvement with different exemplar numbers. We attempt to verify the effectiveness of CLAD by testing it with

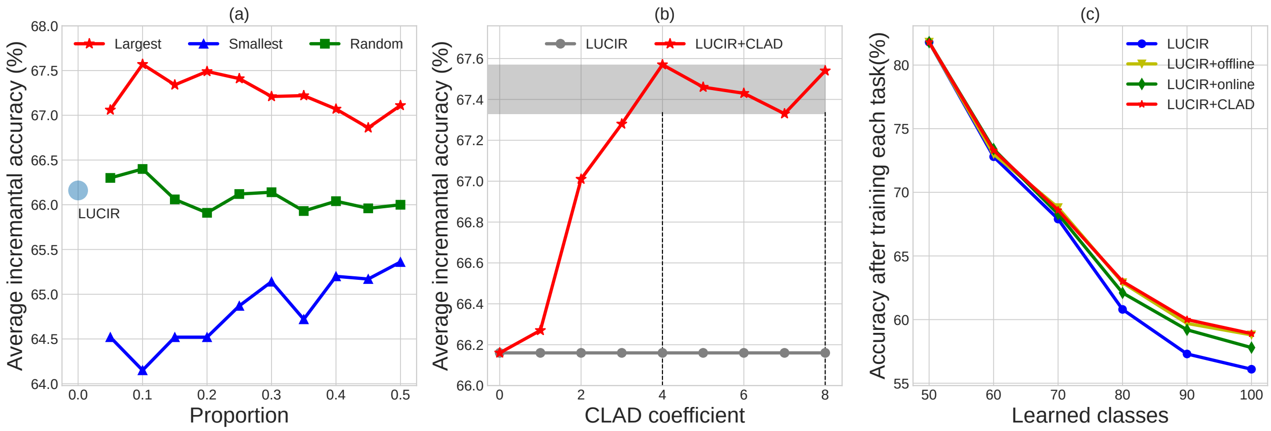

Figure 4: Ablation studies on the effectiveness of conflict prediction (a), the proportion of conflict classes (a), the impact of coefficient of CLAD loss (b), and components in conflict mitigation (c). The average incremental accuracy is reported for each experiment, which is averaged on three runs with different seeds.

R	S=10			S=5		
	LUCIR	$w/$ CLAD	↑	LUCIR	$w/$ CLAD	↑
5	55.65	58.22	**2.57**	52.46	56.09	**3.63**
10	63.28	65.49	2.21	56.80	59.33	2.53
20	66.16	67.57	1.41	60.43	62.15	1.72
30	67.62	68.57	0.95	62.96	63.87	0.91
40	68.58	69.07	0.49	63.51	64.32	0.81

Table 2: Ablation study on the number of exemplars. The number of exemplars per class is denoted by R.

Similarity	LUCIR	*oracle*	logits	cosine
$A_{0.05}(\%)$	66.16	$65.84_{\mathbf{-0.32}}$	$67.06_{\mathbf{+0.90}}$	$67.22_{\mathbf{+1.06}}$
$A_{0.10}(\%)$	66.16	$65.90_{\mathbf{-0.26}}$	$67.57_{\mathbf{+1.38}}$	$67.45_{\mathbf{+1.29}}$

Table 3: Ablation study on the different measurements for forgetting prediction. $A_{0.05}$ and $A_{0.10}$ denote the average incremental accuracy with conflict proportions of 0.05 and 0.10. *logits* is the adopted measurement and *cosine* is provided as an alternative. The result using an oracle model is also given as *oracle*.

varying numbers of exemplars per class. The corresponding results are listed in Tab. 2. Notably, our approach produces increasingly significant performance gains as the number of exemplars decreases. This phenomenon suggests that a reduced number of exemplars per class exacerbates the imbalance of forgetting between old classes, making our approach particularly effective in tackling this challenging scenario.

Measurements of conflict prediction. Various similarity measurements for conflict prediction are available for CLAD. But the logits-based similarity is sufficient for our proposed CLAD. To be more convincing, we compare the performance differences using different similarity measurements in Tab. 3. It shows that there are no obvious differences in performance between the two measurements, and our approach is more concise and efficient. Furthermore, we experimentally prove that FP using the oracle model is ineffective, which supports our aforementioned analysis.

Impact of coefficient of CLAD loss. We demonstrate the improvement with different coefficients of CLAD loss varying from 1.0 to 8.0 in Fig. 4 (b). Interestingly, our method is not sensitive to this coefficient, especially when it is greater than 4.0. This phenomenon indicates that when the current class is dissimilar enough to the old classes in the buffer, a larger coefficient will not have a more significant effect. These results also reflect the robustness of CLAD.

Conclusion

We analyze catastrophic forgetting by revealing imbalanced forgetting in Class Incremental Learning (CIL). Extensive empirical studies and analyses are conducted to establish the connection between imbalanced forgetting and inter-class similarity. Based on this, a forgetting prediction method and a regularization term named CLAD are designed to disentangle the representation interference of similar old and new classes. The effectiveness of CLAD in improving existing methods is demonstrated across multiple experimental settings. Additionally, comprehensive ablation studies are conducted to verify the rationality of our design. This work provides a novel perspective of imbalanced forgetting in CIL, which might stimulate future research in this field.

Limitation. There are also limitations to our CLAD that are worth further exploration. For example, other kinds of losses and old class selection methods may need to be explored. Numerous exemplar-free methods for CIL are not covered in this research. We plan to include them in our future work.

Acknowledgements

This research was supported by the National Natural Science Foundation of China under Grants 62376267, 61976208, 62071466, 62076242, 62222302 and the InnoHK project.

References

Ashok, A.; Joseph, K. J.; and Balasubramanian, V. N. 2022. Class-Incremental Learning with Cross-Space Clustering and Controlled Transfer. *ArXiv*, abs/2208.03767.

Baltruaitis, T.; Ahuja, C.; and Morency, L.-P. 2017. Multimodal Machine Learning: A Survey and Taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41: 423–443.

Buzzega, P.; Boschini, M.; Porrello, A.; Abati, D.; and Calderara, S. 2020. Dark experience for general continual learning: a strong, simple baseline. *Advances in Neural Information Processing Systems*, 33: 15920–15930.

Chaudhry, A.; Ranzato, M.; Rohrbach, M.; and Elhoseiny, M. 2018. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*.

Chen, J.; Nguyen, T.; Gorur, D.; and Chaudhry, A. 2023. Is forgetting less a good inductive bias for forward transfer? *ArXiv*, abs/2303.08207.

Chen, W.; Liu, Y.; Wang, W.; Bakker, E. M.; Georgiou, T.; Fieguth, P. W.; Liu, L.; and Lew, M. S. 2021. Deep Image Retrieval: A Survey. *ArXiv*, abs/2101.11282.

Deng, D.; Chen, G.; Hao, J.; Wang, Q.; and Heng, P.-A. 2021. Flattening Sharpness for Dynamic Gradient Projection Memory Benefits Continual Learning. *Advances in Neural Information Processing Systems*, 34: 18710–18721.

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

Douillard, A.; Cord, M.; Ollion, C.; Robert, T.; and Valle, E. 2020. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *EProceedings of the European Conference on Computer Vision (ECCV)*, 86–102. Springer.

Farajtabar, M.; Azizan, N.; Mott, A.; and Li, A. 2020. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, 3762–3773. PMLR.

He, C.; Wang, R.; and Chen, X. 2021. A tale of two cils: The connections between class incremental learning and class imbalanced learning, and beyond. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3559–3569.

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

Hinton, G.; Vinyals, O.; Dean, J.; et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).

Hou, S.; Pan, X.; Loy, C. C.; Wang, Z.; and Lin, D. 2019. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 831–839.

Hu, X.; Tang, K.; Miao, C.; Hua, X.-S.; and Zhang, H. 2021. Distilling causal effect of data in class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3957–3966.

Krizhevsky, A.; et al. 2009. Learning multiple layers of features from tiny images. In *Technical Report*.

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

Liu, Y.; Schiele, B.; and Sun, Q. 2021. Adaptive aggregation networks for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2544–2553.

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

Lopez-Paz, D.; and Ranzato, M. 2017. Gradient Episodic Memory for Continual Learning. In *Advances in Neural Information Processing Systems*.

McCloskey, M.; and Cohen, N. J. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, 109–165. Elsevier.

Prabhu, A.; Torr, P. H. S.; and Dokania, P. K. 2020. GDumb: A Simple Approach that Questions Our Progress in Continual Learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Ramasesh, V. V.; Dyer, E.; and Raghu, M. 2020. Anatomy of catastrophic forgetting: Hidden representations and task semantics. *arXiv preprint arXiv:2007.07400*.

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

Riemer, M.; Cases, I.; Ajemian, R.; Liu, M.; Rish, I.; Tu, Y.; and Tesauro, G. 2018. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*.

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

Saha, G.; Garg, I.; and Roy, K. 2021. Gradient Projection Memory for Continual Learning. *ArXiv*, abs/2103.09762.

Shi, Y.; Zhou, K.; Liang, J.; Jiang, Z.; Feng, J.; Torr, P. H. S.; Bai, S.; and Tan, V. Y. F. 2022. 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*, 16701–16710.

Van de Ven, G. M.; and Tolias, A. S. 2019. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*.

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

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

Zhao, B.; Xiao, X.; Gan, G.; Zhang, B.; and Xia, S.-T. 2020. Maintaining discrimination and fairness in class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13208–13217.