Dual-Enhanced Coreset Selection with Class-wise Collaboration for Online Blurry Class Incremental Learning

Yutian Luo[1], Shiqi Zhao[2], Haoran Wu[2], Zhiwu Lu[1,*]
[1]Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
[2]China Unicom Research Institute, Beijing, China
{luoyutian2021, luzhiwu}@ruc.edu.cn

Abstract

Traditional online class incremental learning assumes class sets in different tasks are disjoint. However, recent works have shifted towards a more realistic scenario where tasks have shared classes, creating blurred task boundaries. Under this setting, although existing approaches could be directly applied, challenges like data imbalance and varying class-wise data volumes complicate the critical coreset selection used for replay. To tackle these challenges, we introduce DECO (Dual-Enhanced Coreset Selection with Class-wise Collaboration), an approach that starts by establishing a class-wise balanced memory to address data imbalances, followed by a tailored class-wise gradient-based similarity scoring system for refined coreset selection strategies with reasonable score guidance to all classes. DECO is distinguished by two main strategies: (1) Collaborative Diverse Score Guidance that mitigates biased knowledge in less-exposed classes through guidance from well-established classes, simultaneously consolidating the knowledge in the established classes to enhance overall stability. (2) Adaptive Similarity Score Constraint that relaxes constraints between class types, boosting learning plasticity for less-exposed classes and assisting well-established classes in defining clearer boundaries, thereby improving overall plasticity. Overall, DECO helps effectively identify critical coreset samples, improving learning stability and plasticity across all classes. Extensive experiments are conducted on four benchmark datasets to demonstrate the effectiveness and superiority of DECO over other competitors under this online blurry class incremental learning setting.

1. Introduction

Online class incremental learning (OCIL) [25, 37] presents a practical challenge that a model needs to acquire new knowledge while retaining previously learned information, using most of the stream data only once. Recent works [6, 20] point out that traditional OCIL, with disjoint class distribution across tasks, ignores the real-world scenarios where classes are shared across tasks and major classes vary across tasks. To mimic such scenarios, an online blurry class incremental learning (OBCIL) setting [6] is proposed. We show this setting in Figure 1 and follow it for further research.

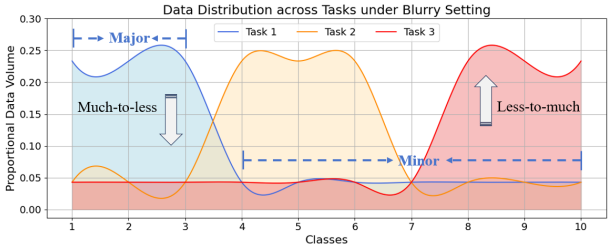


Figure 1. Illustration of the data distribution under blurry setting. All classes are shared and the major classes vary across all tasks.

Among the existing OCIL methods that can be directly applied to the OBCIL scenario, rehearsal-based methods are predominant due to their effective sample selection for replay. Most rehearsal-based methods concentrate on buffer management and buffer usage. For buffer management, the reservoir sampling [11], the mean prototype selection [29], and the coreset selection [34, 39] provide different insights into sampling and updating the buffer. In terms of buffer usage, works vary among selecting the interfered samples [3, 14], adopting contrastive learning [14, 24] as augmentation, and designing new training paradigm [28]. In our work, we follow previous works [6, 28] that adopt the two-stage training paradigm for buffer usage due to their success on OBCIL, but mainly focus on devising a considerable and effective buffer management method.

While the concept of OBCIL was introduced by [6], the critical challenges within this setting remain inadequately addressed. Concretely, we identify two main challenges in OBCIL: **(1) Data imbalance within tasks**, which is exacerbated when replaying with the imbalanced buffer. Although balanced buffers are proposed in some works [6, 29], they often fall short of practical online learning principles due

*Corresponding author.

to the reuse of all stream data [20]. **(2) Varying class-wise data volume across tasks**, which presents two clear patterns. One pattern is a reduction in data volume for classes that were initially major (much-to-less change), while the other involves an increase for classes that become major later on (less-to-much change). The latter, unique to OBCIL and often overlooked, poses a significant challenge. Classes experiencing this increase are less represented initially, resulting in underdeveloped and biased learning. Thus, both buffer updates and model training in later tasks are affected.

In light of the two challenges in the OBCIL setting, we methodically design novel DECO (Dual-Enhanced Coreset Selection with Class-wise Collaboration) to improve model performance. To address data imbalance, we propose a class-wise balanced memory that dynamically adjusts candidate update samples and their classes, maintaining a real-time class-wise balanced buffer. Then, drawing from the successes and drawbacks of recent coreset selection work [39], we devise a class-wise gradient-based scoring system grounded in class balance memory. This system facilitates the assessment of the applicability of each score guidance to the coreset selection set for each class when facing the two challenges and enables tailored coreset selection strategies to meet the unique needs of each class. With this class-wise gradient-based scoring system, we devise two strategies to optimize the score guidance through class-wise collaboration: (1) Collaborative Diverse Score Guidance that mitigates biased knowledge in less-exposed classes through guidance from well-established classes, simultaneously consolidating the knowledge in these established classes to enhance overall stability. (2) Adaptive Similarity Score Constraint that relaxes constraints between class types, boosting learning plasticity for less-exposed classes and assisting well-established classes in defining clearer boundaries, thereby improving overall plasticity. Finally, all these designs form our novel DECO, which selects critical samples for coreset and enhances both stability and plasticity for the model. We compare our DECO with other representative coreset selection and rehearsal-based methods and ablate the effectiveness of each design in DECO. Extensive experiment results on four benchmark datasets show that our DECO reaches the SOTA performance.

Overall, the main contributions of this paper are fourfold: (1) We propose a real-time class-wise balanced memory, CBM, as a new baseline to mitigate data imbalance in the OBCIL setting. (2) We establish a class-wise gradient-based scoring system that facilitates the assessment of diverse score guidance and enables tailored strategies for different classes. (3) We optimize coreset selection with two strategies driven by class-wise collaboration, forming our final method, DECO, which dually enhances the stability and plasticity of the model in all classes. (4) Extensive results demonstrate the superiority of our method.

2. Related Work

2.1. Rehearsal-based OCIL

In the online class incremental learning problem, the rehearsal-based methods show obvious superiority over the other methods due to the buffer replay. Generally, these methods focus on three aspects: buffer management, buffer usage, and additional modifications. Among all works related to buffer management, [11, 30, 36] propose reservoir sampling, which assigns each sample in the buffer the same but gradually decreasing chance of being replaced. In [29], the balanced buffer is set up with the class-wise mean prototypes but only updated after each task. Recent work [6] balances and diversifies the buffer by reusing all stream samples after each task for measurement with the final model. In terms of buffer usage, most works [2, 3, 8, 9, 24] follow the way [11] of combining buffer samples and stream samples in each mini-batch for training. In [28], the buffer is used to retrain a model from scratch for evaluation. In [3, 14], the authors select the maximum interfered buffer samples to train with the stream data. Besides, contrastive learning is also introduced by [14, 24] to diversify all the training samples. In other directions, some methods combine buffer replay with regularized-based constraints like [2, 10, 40] to consolidate the past knowledge. Moreover, some works focus on solving the score bias problem by retraining the model on a balanced subset [38] and adopting split cross-entropy loss for new data [9]. In our work, we focus on buffer management and propose a real-time balanced buffer update way with a dual-enhanced coreset selection method where most stream data only use once.

2.2. Coreset Selection

Coreset selection [7, 15, 17, 18] focuses on identifying the most informative samples from a dataset to create a smaller, representative subset for tasks like classification. Traditional methods in coreset selection primarily deal with the current data. The geometric-based approaches [1, 12, 32, 33] select representative, non-redundant samples based on feature space distances. Besides, the gradient-based methods [19, 26] choose samples that closely approximate the gradients of the entire dataset. Another category is loss-based methods [5, 13, 23, 27, 35], prioritizing samples that contribute most to learning and memorizing dataset knowledge. In addition, some works focus on continual learning scenarios. In [4], the diversity of replay samples is expanded in terms of parameter gradients. In [34], samples with the highest gradient-matching degree are selected and weighted used. Yoon et al. [39] propose different gradient-based similarity scores to select the informative and diverse samples for the OCIL scenario. In this paper, we propose a class-wise scoring system to facilitate the design of diverse coreset selection strategies with class-wise collaboration.

3. Method

3.1. Preliminary

Under the OBCIL setting, we consider the model learns on stream data $\{D_t\}_{t=1}^{N}$, where N denotes the total task number. The samples in each D_t are represented as $\{(x_t^i, y_t^i)\}_{i=1}^{|D_t|}$, where $|D_t|$ denotes the total number of samples in D_t, x_t^i and y_t^i denote the i-th image and label in the t-th task T_t, respectively. Let $\mathcal{C} = \{c_1, c_2, ..., c_K\}$ be the whole set of classes, where K denotes the total number of classes. According to the definition of OBCIL in [6], all the K classes are shared across the T tasks. Thus, we have $\mathcal{C}_t = \mathcal{C}$ $(1 \leq t \leq N)$, where $\mathcal{C}_t$ denotes the set of classes in task T_t. The main difference between tasks lies in the variety of major classes in each $\mathcal{C}_t$. For discussion convenience, we split the $\mathcal{C}_t$ into three subsets and have $\mathcal{C}_t = \{\mathcal{C}_t^P, \mathcal{C}_t^C, \mathcal{C}_t^F\}$, where $\mathcal{C}_t^C$ denotes the current major classes set for task T_t, $\mathcal{C}_t^P$ denotes the past major classes for tasks set $\{T_p\}_{p=1}^{t-1}$, and $\mathcal{C}_t^F$ denotes the future major classes for tasks set $\{T_j\}_{j=t+1}^{N}$. In each task, the samples of current major classes $\mathcal{C}_t^C$ take the dominant ratio of all the $|D_t|$ samples at $(100-Q)\%$, while the samples of past major classes $\mathcal{C}_t^P$ and future major classes $\mathcal{C}_t^F$ take the minor ratio of all the $|D_t|$ samples as $Q\%$. In our experiments, we set Q with 10, 20, and 30 for extensive ablation.

In this paper, we adopt the two-stage training paradigm following [6, 28] for our method and all the other competitors. Let $\{\mathcal{M}_t\}_{t=1}^{N}$ be the set of memory buffer, where $\mathcal{M}_t$ denotes the memory of task T_t. The model learns both on D_t and M_t in the first stage. The objective is written as:

$$\arg\min_{\theta}\Big(\sum_{(x,y)\in D_t} l(f_\theta(x), y) + \sum_{(x,y)\in \mathcal{M}_t} l(f_\theta(x), y)\Big), \quad (1)$$

where θ denotes parameters of the model, $f_\theta(x)$ denote the predicted logits score of x, and $l(\cdot)$ denotes the cross entropy loss. The memory buffer $\mathcal{M}_t$ is only updated before the second stage. In the second stage, the model is only retrained on the fixed memory buffer with the objective:

$$\arg\min_{\theta} \sum_{(x,y)\in \mathcal{M}_t} l(f_\theta(x), y). \quad (2)$$

3.2. Class-wise Balanced Memory

Existing rehearsal-based methods usually follow the classic experience replay [11] method and keep the data distribution in the buffer almost the same as that in the stream data. Under the OBCIL setting, adopting such a buffer management method tends to leave the data and class imbalance in the buffer. The imbalanced buffer reused in the second stage further severe the bias problem when it is replayed, leaving the less-exposed classes with worse initial learning ability.

Although some works [6, 29] try to set up a class-wise balanced buffer, they usually require a review of the stream data once again. We point out that such methods are impractical for online incremental learning. Therefore, we propose a simple real-time class-wise balanced memory (CBM). Concretely, we record the number of saved samples $\{q_{c_i}\}_{i=1}^{K}$ for all the classes, where q_{c_i} denotes the number of saved samples for class c_i. Once the buffer $\mathcal{M}$ is fulfilled, we retrieve the label of stream data and compare the corresponding q_{c_i} with the average buffer size $\frac{m}{K}$ for each class in $\mathcal{M}$. If the arriving stream data is of the less-than-average class, we randomly replace a sample from one of the more-than-average classes with it. In reverse, we apply the class-wise reservoir (CW_RSV) update within its class. The pseudocode is presented in Algorithm 1 and the detail of the class-wise reservoir is in Algorithm 2 in the supplementary material. With the CBM, the buffer can easily keep real-time balance among classes and reach a better performance than experienced replay [11] under OBCIL.

3.3. Class-wise Scoring System

Although the CBM achieves real-time class-wise balance, it relies on the suboptimal random sample replacement. Considering data imbalances and varying class volumes, a more complete system to evaluate and select samples in each class is essential. In recent coreset selection work [39], gradient-based similarity score ($\mathcal{S}$), diversity score ($\mathcal{V}$), and affinity score ($\mathcal{A}$) are proposed to assess sample importance. However, $\mathcal{S}$ and $\mathcal{V}$, calculated within mini-batches or batch candidates, are skewed by class imbalances. In addition, the limited ranking scope of these three scores to current task data overlooks updates to previously saved coreset samples, leading to ineffective class-wise assessments. To address these, we introduce a class-wise scoring system based on CBM that thoroughly evaluates all samples in each class.

We denote the class-wise gradient-based similarity score as $\mathcal{CW}\_\mathcal{S}$. When a stream data $(x, y) \in \mathcal{B}_t$ arrives, we identify its target class c_k via CBM, then calculate scores for each sample $\boldsymbol{s}_j^{c_k} \in \mathcal{M}_t^{c_k}$, where $1 \leq j \leq |\mathcal{M}_t^{c_k}|$ and $\mathcal{M}_t^{c_k} \subset \mathcal{M}_t$. For stream data in the more-than-average class, c_k is its own class and it is added to $\mathcal{M}_t^{c_k}$ before the score calculation. Mathematically, $\mathcal{CW}\_\mathcal{S}$ is written as:

$$\mathcal{CW}\_\mathcal{S}(\boldsymbol{s}_j^{c_k}|\mathcal{M}_t^{c_k}) = \frac{\mathcal{G}(\boldsymbol{s}_j^{c_k})\overline{\mathcal{G}(\mathcal{M}_t^{c_k})}^{\top}}{\|\mathcal{G}(\boldsymbol{s}_j^{c_k})\| \cdot \|\overline{\mathcal{G}(\mathcal{M}_t^{c_k})}\|}, \quad (3)$$

where $\mathcal{G}(\boldsymbol{s}_j^{c_k})$ denotes the function of retrieving the gradient vector of $\boldsymbol{s}_j^{c_k}$ in fc layer, $\overline{\mathcal{G}(\mathcal{M}_t^{c_k})}$ denotes the average gradient vector of samples in $\mathcal{M}_t^{c_k}$. Similarly, we denote the class-wise diversity score as $\mathcal{CW}\_\mathcal{V}$, which is written as:

$$\mathcal{CW}\_\mathcal{V}(\boldsymbol{s}_j^{c_k}|\overline{\mathcal{M}_t^{c_k}}) = -\frac{1}{N_k}\sum_{p\neq j}^{N_k} \frac{\mathcal{G}(\boldsymbol{s}_j^{c_k})\mathcal{G}(\boldsymbol{s}_p^{c_k})^{\top}}{\|\mathcal{G}(\boldsymbol{s}_j^{c_k})\| \cdot \|\mathcal{G}(\boldsymbol{s}_p^{c_k})\|}, \quad (4)$$

where $\boldsymbol{s}_p^{c_k} \in \overline{\mathcal{M}_t^{c_k}}$ denotes the other samples in $\mathcal{M}_t^{c_k}$ except $\boldsymbol{s}_j^{c_k}$ and $N_k = |\mathcal{M}_t^{c_k}| - 1$.

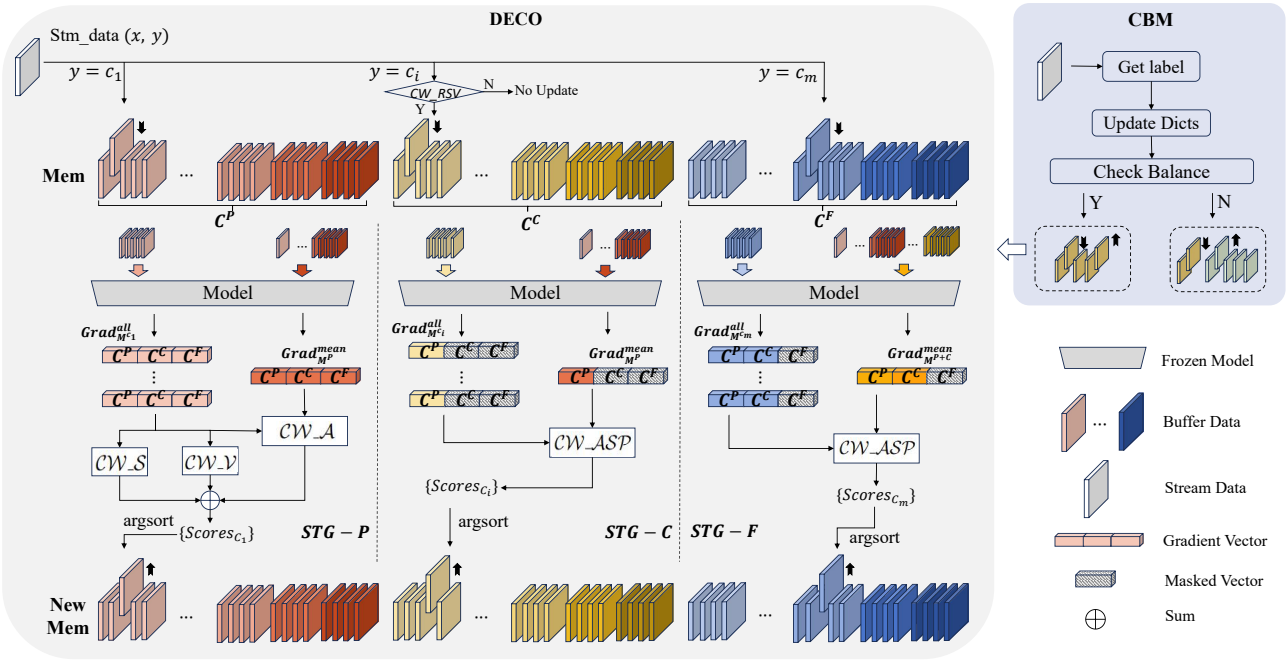


Figure 2. Overview of the DECO. DECO is based on CBM and optimizes the random buffer update in CBM. DECO adopts the DSG and ASC strategies for score calculation of different classes and finally updates the corresponding coreset by removing the data with the lowest score. We illustrate three cases where the target class is $c_1 \in \mathcal{C}^P$, $c_i \in \mathcal{C}^C$, and $c_m \in \mathcal{C}^F$ in this figure. We here only show the situation that the buffer has reached a balanced state. In an unbalanced state, DECO adopts no CW_RSV to remove the lowest-score sample.

When calculating the class-wise affinity score, $\mathcal{CW}\_\mathcal{A}$, we first consider the guidance from previously accumulated knowledge in each class. There's a concern that in less-exposed classes, the coreset samples might not accurately represent the true class distribution, potentially leading to misleading guidance. To address the concern, we introduce a ranking system for affinity guidance that prioritizes classes with more reliable coreset samples. As detailed in Section 3.1, we distinguish the classes in task T_t into three groups based on the variation of major classes and their exposure level: $\mathcal{C}_t^P$, $\mathcal{C}_t^C$, and $\mathcal{C}_t^F$. We hypothesize that the reliability of the coreset samples for providing affinity guidance decreases from $\mathcal{C}_t^P$ to $\mathcal{C}_t^F$. Thus, the system assigns higher-ranking guidance to classes where the coreset samples are deemed more representative of their respective classes. Mathematically, the system defines $\mathcal{CW}\_\mathcal{A}$ as:

$$\mathcal{CW}\_\mathcal{A}(\boldsymbol{s}_j^{c_k}|\mathcal{M}_t^\Omega) = \frac{\mathcal{G}(\boldsymbol{s}_j^{c_k})\overline{\mathcal{G}(\mathcal{M}_t^\Omega)}^\top}{\|\mathcal{G}(\boldsymbol{s}_j^{c_k})\| \cdot \|\overline{\mathcal{G}(\mathcal{M}_t^\Omega)}\|}, \quad (5)$$

where $\Omega = \begin{cases} \mathcal{C}_t^P \cup \mathcal{C}_t^C, & \text{if } c_k \in \mathcal{C}_t^F \\ \mathcal{C}_t^P, & \text{otherwise} \end{cases}$. Note that in task T_1 there is no $\mathcal{C}_1^P$, so we only calculate the $\mathcal{CW}\_\mathcal{S}$ and $\mathcal{CW}\_\mathcal{V}$ for sample selection in c_k when $c_k \in \mathcal{C}_1^C$ in task T_1.

To distinguish the classes into $\mathcal{C}_t^P$, $\mathcal{C}_t^C$, and $\mathcal{C}_t^F$ without previous knowledge of class distribution in the current task. We use one dictionary to record the total samples encountered for each class, categorizing those with above-average counts into $\mathcal{C}_t^P$ after every task, and another monitors the class-wise sample count within each task, identifying the classes belonging to $\mathcal{C}_t^C$; the remaining classes fall into $\mathcal{C}_t^F$.

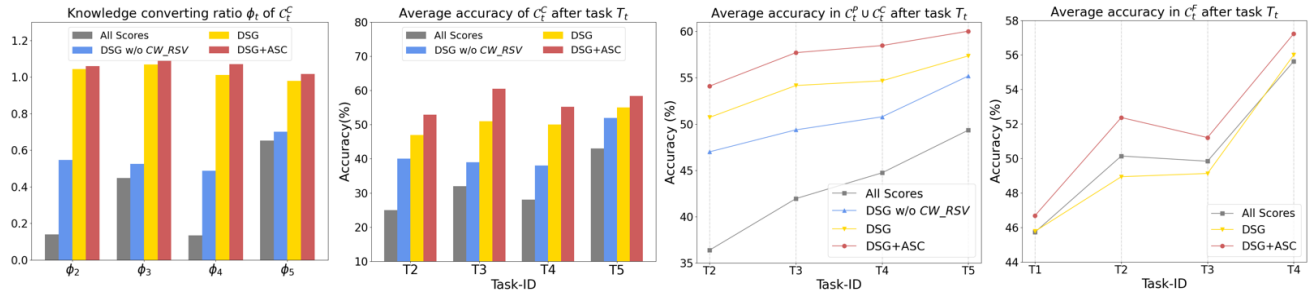
With all these class-wise gradient-based scores, we complete the class-wise scoring system which facilitates the assessment of score guidance in each class, enabling the development of tailored coreset selection strategies that are more responsive to the unique needs of each class.

3.4. Coreset Selection with Diverse Score Guidance

In our class-wise scoring system, the $\mathcal{CW}\_\mathcal{S}$ and the $\mathcal{CW}\_\mathcal{V}$ provide intra-class scores guidance for coreset selection. This means the guidance's effectiveness largely depends on the class's learning progress and the quality of its existing coreset samples. For well-established classes in $\mathcal{C}_t^P$, which are already familiar as past major classes, the intra-class score guidance is reliable. Therefore, we adopt all three scores for coreset selection strategy in $\mathcal{C}_t^P$, denoted as:

$$STG - P = \{\mathcal{CW}\_\mathcal{S} + \mathcal{CW}\_\mathcal{V} + \mathcal{CW}\_\mathcal{A}\} \quad (6)$$

Differently, the score guidance provided by of $\mathcal{CW}\_\mathcal{S}$ and $\mathcal{CW}\_\mathcal{V}$ may be doubtful for classes in $\mathcal{C}_t^C$ and $\mathcal{C}_t^F$. For the classes in $\mathcal{C}_t^F$, the class exposure is lacking. Besides, the distribution of seen samples is possibly biased from the real distribution and inconsistent across tasks. Therefore, the initially learned knowledge for these classes is possibly



(a) (b) (c) (d)

Figure 3. Validation results of adopting diverse score guidance and the adaptive score constraint."All score" denotes always adopting all class-wise scores. We show the results after task T_t in: **(a)** knowledge converting ratio ϕ_t, **(b)** average accuracy of $\mathcal{C}_t^C$, **(c)** average accuracy change of $\mathcal{C}_t^P \cup \mathcal{C}_t^C$, and **(d)** average accuracy change of $\mathcal{C}_t^F$ on CIFAR10 to validate the enhanced stability and plasticity.

poor and the intra-class guidance is unreliable and inconsistent across tasks. Therefore, we abandon the intra-class guidance scores $\mathcal{CW}\_\mathcal{S}$ and $\mathcal{CW}\_\mathcal{V}$, only adopting $\mathcal{CW}\_\mathcal{A}$ to guide the coreset selection in $\mathcal{C}_t^F$. The strategy of coreset selection for classes in $\mathcal{C}_t^F$ can be denoted as:

$$STG-F = \{\mathcal{CW}\_\mathcal{A}\} \tag{7}$$

The reasons behind this modification are (1) The score guidance from well-exposed classes in $\mathcal{C}_t^P$ and $\mathcal{C}_t^C$ with $\mathcal{CW}\_\mathcal{A}$ is more reasonable and consistent. Thus, the learning in $\mathcal{C}_t^F$ is more effective and the learned knowledge, as a good initialization, can be well-retained and utilized by later learning. (2) With $\mathcal{CW}\_\mathcal{A}$ as the only coreset selection guidance for $\mathcal{C}_t^F$, the coreset samples in $\mathcal{C}_t^F$ are more beneficial to the old knowledge retention in $\mathcal{C}_t^P$. Therefore, such class-wise collaboration enhances the stability in all classes.

For $\mathcal{C}_t^C$ classes, which are less exposed before task T_t but have numerous samples in T_t, we combine the CW_RSV with $\mathcal{CW}\_\mathcal{A}$ for coreset updates. This strategy ensures a gradual update process, aiding in solidifying the model's knowledge in $\mathcal{C}_t^C$. The coreset selection strategy for classes in $\mathcal{C}_t^C$ can be written as:

$$STG-C = \{CW\_RSV + \mathcal{CW}\_\mathcal{A}\} \tag{8}$$

Note that there are two special cases for this strategy: First, in task T_1, without the guidance of well-established classes, we only use CW_RSV for $STG-C$. Second, when we need to balance the buffer by removing a sample in $\mathcal{C}_t^C$, we only adopt $\mathcal{CW}\_\mathcal{A}$ for $STG-C$ to find the sample with the lowest score. Overall, all these three strategies compose our diverse score guidance **(DSG)** for coreset selection.

To validate the effectiveness of each detail in our design, and to confirm our analytical reasoning, we first introduce the knowledge converting ratio ϕ_t which is calculated as:

$$\phi_t = \frac{Perf(\mathcal{C}_t^C, t)}{Perf(\mathcal{C}_t^C, t-1)}, \tag{9}$$

where $t > 1$ and $Perf(\mathcal{C}_t^C, t)$ represents the model's performance on the classes in $\mathcal{C}_t^C$ after task T_t. It measures the performance change in $\mathcal{C}_t^C$ between before and after task T_t. We assume that ϕ_t is positively related to the quality of the learned knowledge of these classes saved in the less-exposed stages. In Figure 3 (a), only adopting $\mathcal{CW}\_\mathcal{A}$ as score guidance achieves higher ϕ_t than all scores strategy, which validates the improved knowledge quality and learning consistency in less-exposed classes. Besides, combining CW_RSV further improves the ϕ_t which validates the effectiveness of CW_RSV in assisting learning on plenty of stream data. For direct comparison, we also show the accuracy of $\mathcal{C}_t^C$ with different strategies just after task T_t in Figure 3 (b), which again verifies their effectiveness. In Figure 3 (c), the stability enhancement in well-established classes ($\mathcal{C}_t^P \cup \mathcal{C}_t^C$) is also validated by the higher accuracy when adopting only $\mathcal{CW}\_\mathcal{A}$ and combining CW_RSV. Overall, the DSG is validated to effectively enhance the model's stability across all classes. More ablation results within DSG are provided in the supplement.

3.5. Selection with Adaptive Score Constraint

In our class-wise scoring system, the affinity similarity in $\mathcal{CW}\_\mathcal{A}(\boldsymbol{s}_j^{c_k}|\mathcal{M}_t^\Omega)$ is calculated using the full gradient vector of the FC classifier layer between samples in $\mathcal{M}_t^\Omega$ and $\boldsymbol{s}_j^{c_k}$. However, the affinity measurement with such a full gradient vector may not be optimal. Considering the exposed degree of different classes, we divide this full gradient vector into two parts which correspond to classes in Ω and that in the rest $\mathcal{C}\backslash\Omega$, which are retrieved by $\mathcal{G}^\Omega(\cdot)$ and $\mathcal{G}^{\mathcal{C}\backslash\Omega}(\cdot)$.

Since the classes in $\mathcal{C}\backslash\Omega$ are less-exposed, we suppose the gradient guidance from $\mathcal{G}^{\mathcal{C}\backslash\Omega}(\cdot)$ contributes less to knowledge consolidation, and calculating affinity score with this part may also hinder the plasticity in $\mathcal{C}\backslash\Omega$. To address this, we introduce a split class-wise affinity score, $\mathcal{CW}\_\mathcal{ASP}$, with adaptive gradient score constraint **(ASC)**:

$$\mathcal{CW}\_\mathcal{ASP}(\boldsymbol{s}_j^{c_k}|\mathcal{M}_t^\Omega) = \frac{\mathcal{G}^\Omega(\boldsymbol{s}_j^{c_k})\overline{\mathcal{G}^\Omega(\mathcal{M}_t^\Omega)}^\top}{\|\mathcal{G}^\Omega(\boldsymbol{s}_j^{c_k})\| \cdot \|\overline{\mathcal{G}^\Omega(\mathcal{M}_t^\Omega)}\|} \tag{10}$$

This relaxed gradient constraint continues the guidance from Ω to the coreset selection of $\mathcal{C}\backslash\Omega$ while also enabling

the learning plasticity in $\mathcal{C}\backslash\Omega$, leading to a more diverse coreset. This diverse coreset in turn helps classes in Ω establish better boundaries, enhancing the model's overall plasticity. In Figure 3 (d), the effectiveness of ASC in enhancing plasticity in $\mathcal{C}_t^F$ is verified with the improved accuracy. Besides, the collaboratively improved knowledge quality and the learning plasticity in previous classes are demonstrated by the improved (highest) performance shown in Figure 3 (a)-(c), respectively. Overall, we denote the final class-wise coreset selection strategy, the core idea of DECO, as:

$$STG-X=\begin{cases}\{\mathcal{CW}\_\mathcal{S}+\mathcal{CW}\_\mathcal{V}+\mathcal{CW}\_\mathcal{A}\}, & X=P\\ \{CW\_RSV+\mathcal{CW}\_\mathcal{ASP}\}, & X=C\\ \{\mathcal{CW}\_\mathcal{ASP}\}, & X=F\end{cases} \quad (11)$$

With this comprehensive strategy, the coreset selection is guided by a class-wise collaboration score. As shown in Figure 2, we remove the sample in the target class with the lowest score to get new coreset for replay in later training.

4. Experiment

4.1. Datasets and Metrics

Datasets. Following previous work [6], we evaluate our method and other competitor methods on four commonly used benchmark datasets: MNIST [22], CIFAR10 [21], CIFAR100 [21], and ImageNet [31]. Following [4, 6], we denote the blurry setup as "Blurry$Q$" where Q indicates the portion of the samples in shared classes across all the tasks. In each task, the major classes samples account for $(1-Q)\%$ and the shared classes samples account for $Q\%$. The class distribution of shared classes is balanced in each task. We split the datasets following the way in [6]. In MNIST and CIFAR10, we split them into 5 tasks with 2 major classes in each task. In CIFAR100 and ImageNet, we split them into 10 tasks with 10 and 100 major classes in each task, respectively. More details can be referred to [6].
Metrics. Following [6], we adopt the Final Average Accuracy and Final Forgetting as the metric. Assuming a_i^t as the top-1 accuracy of model M_t on the i-th task after training on task T_t, the FAA is formally written as:

$$\text{FAA}=\frac{1}{n}\sum_{i=1}^{n}a_i^n, \quad (12)$$

where n denotes the total number of tasks. The Final Forgetting indicates the forgetting on previous $(n-1)$ tasks of the final model. Under the OBCIL, the forgetting refers to the gap between the best performance and the final performance of the model on the major classes of each task during the whole training. the FF is formally written as:

$$\text{FF}=\frac{1}{n-1}\sum_{j=1}^{n-1}f_j, \quad \text{s.t. } f_j=\max_{l\in\{1,\dots,n-1\}}\mu_j^l-\mu_j^{n-1}, \quad (13)$$

where μ_j^l denotes the accuracy on $\mathcal{C}_j^C$ in task T_l.

4.2. Baselines and Implementation Details

We compare our method with various competitive rehearsal-based methods. In terms of buffer management, we adopt ICARL [29], ER [11], Gdumb [28], OCS [39], and RM [6] as the competitors, where OCS is the representative gradient-based coreset selection method. Note that although RM takes advantage of the reuse of all the stream data [20] for buffer selection after each task and is thereby impractical under the online setting, we still include it in the final comparison to illustrate the superiority of our method. Besides, we also adopt strong baselines related to buffer usage like MIR [3], SCR [24], and DVC [14], and those related to bias correction like BIC [38] and ER-ACE [8].

We adopt the same network architecture as [6], where MLP400, ResNet18, Resnet32 and ResNet34 [16] are adopted for MNIST, CIFAR10, CIFAR100, and ImageNet, respectively. For DVC, we add Q-net in network architecture following the original requirement. For other hyperparameters, we set the batch size as 16 on MNIST, CIFAR10, and CIFAR100 which is the same with [6] and the batch size of 128 on ImageNet. We adopt the cosine annealing learning rate from 0.0005 to 0.5 on MNIST, CIFAR10, and CIFAR100, and from 0.001 to 0.1 on ImageNet. The total epoch in the second stage is 256 for all methods. All these are the same with [6, 28]. We ran the experiments on three randomly generated tasks split on MNIST and CIFAR10/100 and one task split on ImageNet, following [6]. For reproduction, We adopt the public codes provided by [6] for ICARL, BIC, Gdumb, and RM, and reproduce the other methods under the OBCIL setting with their official codes.

4.3. Main results

For a clear comparison, we first show the upper bound (joint training) results on each dataset in Table 1. Further, we compare the results of all competitors and our proposed method under the 'Blurry10' setting for overall evaluation. From Table 1, we have the following observations.

In terms of final average accuracy, our proposed DECO outperforms all the other methods on all the four datasets. The performance gap among these methods becomes more obvious on CIFAR10 and ImageNet, where our DECO outperforms all the other methods by 3.08% and 1.45% under fair comparison (*i.e.*, except RM), respectively. Particularly, our DECO even beats the strongest competitor RM regardless of their reuse of the stream data. On MNIST and CIFAR100, although the performance gap among methods is smaller, our method still has the leading performance than any other method by almost 1%. Note that compared to all the gradient-based coreset selection methods, our method shows a remarkable difference over all the datasets under the OBCIL setting. All these observations provide direct evidence that our proposed DECO is effective in solving the special challenges in this setting and successfully enhances

Table 1. Comparative results on Final Average Accuracy (FAA) and Final Forgetting (FF) between our DECO and other competitor methods. $|M|$ denotes the buffer size. The method with † takes advantage of reusing stream data for buffer sample selection.

Methods	**MNIST** ($\|M\|$=500)		**CIFAR10** ($\|M\|$=500)		**CIFAR100** ($\|M\|$=2000)		**ImageNet** ($\|M\|$=20000)	
Upper bound	98.39		95.05		72.21		89.64	
Buffer size	FAA ↑	FF ↓	FAA ↑	FF ↓	FAA ↑	FF ↓	FAA ↑	FF ↓
ICARL [29]	78.26 ±0.45	6.70 ±0.37	45.31 ±2.89	4.75 ±1.92	17.84 ±0.78	5.38 ±0.55	17.52	1.94
BIC [38]	77.62 ±1.47	7.95 ±1.05	42.06 ±2.03	1.34 ±2.35	13.21 ±0.19	4.16 ±0.33	37.59	1.83
Gdumb [28]	88.66 ±0.51	2.31 ±0.30	49.41 ±0.69	1.47 ±2.02	26.58 ±0.56	7.17 ±0.70	21.52	4.07
ER [11]	88.06 ±0.70	9.20 ±0.23	52.60 ±2.43	19.57 ±3.71	32.08 ±1.86	12.55 ±1.57	34.37	17.28
MIR [3]	88.76 ±0.62	6.30 ±0.41	54.03 ±2.82	14.40 ±1.97	33.06 ±0.83	13.58 ±0.95	37.21	14.56
SCR [24]	88.82 ±0.88	9.33 ±0.55	53.37 ±2.77	16.37 ±1.31	32.78 ±0.60	15.26 ±1.12	38.18	13.83
OCS [39]	89.12 ±0.37	5.90 ±0.85	55.37 ±1.26	13.15 ±0.87	32.48 ±0.89	12.13 ±0.64	38.99	12.97
ER-ACE [9]	89.25 ±0.53	6.20 ±1.03	56.08 ±2.99	12.73 ±1.02	32.83 ±1.27	11.03 ±1.30	41.13	12.80
DVC [14]	88.90 ±0.65	6.93 ±0.92	56.95 ±1.07	11.20 ±1.92	32.98 ±1.02	13.97 ±0.82	41.11	11.28
RM† [6]	90.24 ±0.64	1.02 ±1.17	58.14 ±1.96	-0.12 ±0.42	32.89 ±1.22	3.59 ±0.53	41.29	1.41
DECO	**90.89 ±0.53**	**0.72 ±0.93**	**60.03 ±1.37**	**-1.96 ±1.15**	**33.93 ±0.71**	**3.35 ±0.66**	**42.58**	**1.03**

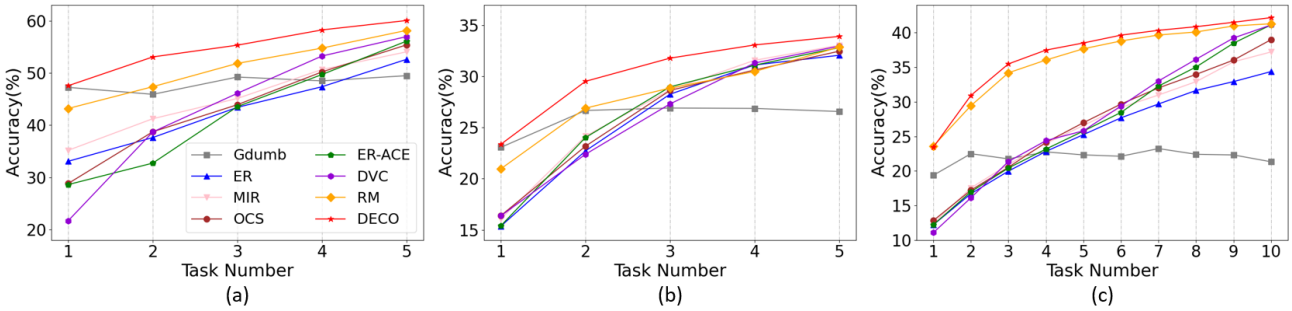


Figure 4. Visualized results of accuracy changes of the model during the sequential training under online Blurry10 setting on (a) CIFAR10, (b) CIFAR100, (c) ImageNet. We only select the results of some competitive methods and our DECO for comparison.

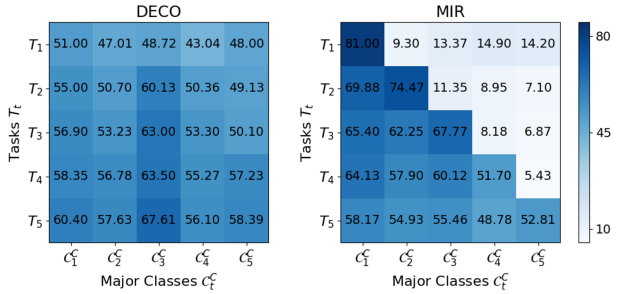


Figure 5. Comparison between DECO and MIR on the average accuracy of the model in task-wise major classes on CIFAR10.

the stability and plasticity of the learned model.

In terms of the final forgetting, the methods with a balanced memory buffer (*i.e.*, ICARL, BIC, Gdumb, RM, and our DECO) show much better performance in mitigating the catastrophic forgetting problem. Among these methods, our DECO still shows obvious advantages on all datasets and even achieves negative final forgetting. We point out that it is because our DECO help model continually learns in each class with enhanced stability and plasticity. To directly illustrate the huge difference in final forgetting, we show the task-wise results on the major classes $\mathcal{C}_t^C$ of each task T_t on CIFAR10 in Figure 5 for an example. Although in early tasks (*i.e.*, T_{1-3}), MIR reaches higher initial results on the corresponding major classes (*i.e.*, $\mathcal{C}_{1-3}^C$), the performance on these results quickly drops in the latter tasks and finally fall behind the performance of our methods on all the major classes. In reverse, we can observe that our DECO, though achieves relatively lower initial performance for the early major classes, keeps helping the model learn in these classes and the performance in all the classes continually improves during the training and finally reaches higher accuracy in all classes than MIR does. Besides, it can be observed that our DECO also provides reasonable guidance for the less-exposed future major classes in the early stage and this indeed helps the model learn consistent knowledge in these classes, which benefits the learning in later tasks. For more task-wise performance of the other methods, please refer to the supplementary material.

In addition to the final performance of the methods, we also show the changing average performance on CIFAR10, CIFAR100, and ImageNet throughout the whole training in Figure 4. It could be observed that our DECO consistently reaches the best performance on each of the datasets. Moreover, our DECO keeps the best performance throughout the whole training. All these observations again validate that DECO enhances the stability and plasticity of the model.

Overall, all these comparisons provide evidence of the superiority of our DECO as an effective coreset selection

Table 2. Ablation results on Final Average Accuracy (FAA) and Final Forgetting (FF) by combining different strategies in our DECO. "CBM", "DSG", and "ASC" denote class-wise balanced memory, diverse score guidance, and adaptive score constraint, respectively.

Strategies			MNIST		CIFAR10		CIFAR100		ImageNet	
CBM	**DSG**	**ASC**	FAA ↑	FF ↓	FAA ↑	FF ↓	FAA ↑	FF ↓	FAA ↑	FF ↓
			88.06 ±0.70	9.20 ±0.23	52.60 ±2.43	19.57 ±3.71	32.08 ±1.86	12.55 ±1.57	37.15	16.28
✓			89.71 ±0.67	2.12 ±1.05	56.19 ±1.21	0.77 ±0.85	32.18 ±1.09	6.57 ±0.80	37.60	6.01
✓	✓		90.20 ±0.47	1.17 ±0.68	57.36 ±0.92	0.51 ±1.03	33.07 ±1.22	4.40 ±0.67	41.13	2.01
✓	✓	✓	**90.89 ±0.53**	**0.72 ±0.93**	**60.03 ±1.37**	**-1.96 ±1.15**	**33.93 ±0.71**	**3.35 ±0.66**	**42.58**	**1.03**

Table 3. Average ablation results of our method and other competitors with different buffer sizes on CIFAR10.

	$\|M\| = 200$		$\|M\| = 500$		$\|M\| = 1000$	
Methods	FAA ↑	FF ↓	FAA ↑	FF ↓	FAA ↑	FF ↓
Gdumb [28]	35.85	1.67	49.47	1.44	64.26	1.12
ER [11]	38.18	22.05	52.60	19.57	66.58	13.71
MIR [3]	39.80	21.30	54.03	14.40	67.75	12.55
OCS [39]	40.35	20.17	55.37	13.15	68.02	13.19
ER-ACE [9]	38.13	18.03	56.08	12.73	68.04	13.28
DVC [14]	41.17	19.95	56.95	11.20	68.29	14.72
RM† [6]	44.01	0.90	58.14	-0.12	68.78	-1.09
DECO	**45.61**	**-0.31**	**60.03**	**-1.96**	**69.07**	**-1.23**

Table 4. Average ablation results of our method and other competitors under different blurry extent settings on CIFAR10.

	Blurry10		**Blurry20**		**Blurry30**	
Methods	FAA ↑	FF ↓	FAA ↑	FF ↓	FAA ↑	FF ↓
Gdumb [28]	49.47	1.44	48.82	0.88	47.78	1.37
ER [11]	52.60	19.57	54.81	15.54	53.04	17.53
MIR [3]	54.03	14.40	54.96	14.02	53.87	17.98
OCS [39]	55.37	13.15	56.07	14.31	55.81	13.68
ER-ACE [9]	56.08	12.73	57.19	15.02	56.53	12.22
DVC [14]	56.95	11.20	58.15	14.73	56.09	13.31
RM† [6]	58.14	-0.12	58.71	-0.52	59.82	2.97
DECO	**60.03**	**-1.96**	**60.13**	**-0.95**	**60.66**	**1.11**

method for online continual learning. It enhances the model both in stability and plasticity to finally achieve state-of-the-art performance under the OBCIL setting.

4.4. Ablation Study

Our proposed DECO is composed of three strategies: (1) A real-time class-wise balanced memory buffer CBM. (2) Stability-enhanced class-wise collaboration with diverse score guidance (DSG) for coreset selection. (3) Plasticity-enhanced class-wise collaboration with adaptive score constraint (ASC) for coreset selection. In Table 2, we ablate on these three strategies to prove their effectiveness. Since a balanced memory is crucial and fundamental for class-wise coreset selection, we validate the latter strategies with the CBM adopted. We can observe that: (1) Compared with reservoir coreset selection [11], adopting CBM improves the FAA on all datasets and greatly decreases the final forgetting degree. That validates the necessity of a real-time balanced memory buffer and its effectiveness in mitigating the scores bias [8] which is one of the main reasons for the high final forgetting (This can also be observed in Figure 5). (2) With the DSG strategy adopted, we find further improvements in the FAA and decreases in FF, which indicate that our diverse score guidance achieves success in enhancing the stability of the model. (4) By adopting ASC, the final performance again achieves improvement and all the combined strategies help the model reach the highest FAA and the lowest FF, which proves that all strategies can collaborate well for consistent final improvement. Overall, all these observations provide evidence for the effectiveness and rationality of our proposed method.

In addition, we also ablate on the buffer size and the blurry extent of the setting. To ablate on the buffer size, we compare the performances of all methods with different buffer sizes of 200, 500, and 1000 on CIFAR10. (The ablation results on other datasets are provided in the supplementary material.) In Table 3, we can observe that our DECO always keeps the leading performance regardless of the variation of the buffer size. To ablate the blurry degree of the setting, we set the ratio of the shared class data as 10%, 20%, and 30% and show the results on CIFAR10 in Table 4. It can be observed that our DECO consistently reaches the best performance under the settings of different blurry extents. Overall, all these observations provide evidence that our method keeps its superiority regardless of the variation of key hyperparameters and changes in setting, which further proves the effectiveness of our method.

Moreover, we also conduct experiments with different augmentation strategies applied to all methods to find that our method still reaches the best performance. Please refer to the supplementary material for detailed results.

5. Conclusion

In this paper, we focus on exploring the OBCIL setting. We first identify the two key challenges in this setting: data imbalance and varying class-wise data volume. To address them, we propose a novel dual-enhanced coreset selection method called DECO. It has two novel components, i.e., class-wise collaboration with devised diverse score guidance and adaptive score constraint strategies, to enhance both stability and plasticity across all classes. Finally, we conduct extensive experiments to show the superiority of DECO over other methods under the OBCIL setting.

Acknowledgements This work was supported by National Natural Science Foundation of China (62376274).

References

[1] Sharat Agarwal, Himanshu Arora, Saket Anand, and Chetan Arora. Contextual diversity for active learning. In *European Conference on Computer Vision*, pages 137–153. Springer, 2020. 2

[2] Hongjoon Ahn, Jihwan Kwak, Subin Lim, Hyeonsu Bang, Hyojun Kim, and Taesup Moon. Ss-il: Separated softmax for incremental learning. In *IEEE/CVF International Conference on Computer Vision*, pages 844–853, 2021. 2

[3] Rahaf Aljundi, Lucas Caccia, Eugene Belilovsky, Massimo Caccia, Min Lin, Laurent Charlin, and Tinne Tuytelaars. Online continual learning with maximally interfered retrieval. *ArXiv*, abs/1908.04742, 2019. 1, 2, 6, 7, 8

[4] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *Advances in Neural Information Processing Systems*, 32, 2019. 2, 6

[5] Olivier Bachem, Mario Lucic, and Andreas Krause. Coresets for nonparametric estimation-the case of dp-means. In *International Conference on Machine Learning*, pages 209–217. PMLR, 2015. 2

[6] Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8218–8227, 2021. 1, 2, 3, 6, 7, 8

[7] Zalán Borsos, Mojmir Mutny, and Andreas Krause. Coresets via bilevel optimization for continual learning and streaming. *Advances in Neural Information Processing Systems*, 33:14879–14890, 2020. 2

[8] Matteo Boschini, Lorenzo Bonicelli, Pietro Buzzega, Angelo Porrello, and Simone Calderara. Class-incremental continual learning into the extended der-verse. *arXiv preprint arXiv:2201.00766*, 2022. 2, 6, 8

[9] Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. *arXiv preprint arXiv:2104.05025*, 2021. 2, 7, 8

[10] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *European Conference on Computer Vision*, pages 233–248, 2018. 2

[11] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019. 1, 2, 3, 6, 7, 8

[12] Yutian Chen, Max Welling, and Alex Smola. Super-samples from kernel herding. *arXiv preprint arXiv:1203.3472*, 2012. 2

[13] Sanjoy Dasgupta, Daniel Hsu, Stefanos Poulis, and Xiaojin Zhu. Teaching a black-box learner. In *International Conference on Machine Learning*, pages 1547–1555. PMLR, 2019. 2

[14] Yanan Gu, Xu Yang, Kun Wei, and Cheng Deng. Not just selection, but exploration: Online class-incremental continual learning via dual view consistency. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7442–7451, 2022. 1, 2, 6, 7, 8

[15] Chengcheng Guo, Bo Zhao, and Yanbing Bai. Deepcore: A comprehensive library for coreset selection in deep learning. In *International Conference on Database and Expert Systems Applications*, pages 181–195. Springer, 2022. 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 6

[17] Tyler B Johnson and Carlos Guestrin. Training deep models faster with robust, approximate importance sampling. *Advances in Neural Information Processing Systems*, 31, 2018. 2

[18] Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International Conference on Machine learning*, pages 2525–2534. PMLR, 2018. 2

[19] Krishnateja Killamsetty, Sivasubramanian Durga, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training. In *International Conference on Machine Learning*, pages 5464–5474. PMLR, 2021. 2

[20] Hyunseo Koh, Dahyun Kim, Jung-Woo Ha, and Jonghyun Choi. Online continual learning on class incremental blurry task configuration with anytime inference. *arXiv preprint arXiv:2110.10031*, 2021. 1, 2, 6

[21] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*, 2009. 6

[22] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 6

[23] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pages 6781–6792. PMLR, 2021. 2

[24] Zheda Mai, Ruiwen Li, Hyunwoo Kim, and Scott Sanner. Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3589–3599, 2021. 1, 2, 6, 7

[25] Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469:28–51, 2022. 1

[26] Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pages 6950–6960. PMLR, 2020. 2

[27] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Advances in Neural Information Processing Systems*, 34:20596–20607, 2021. 2

[28] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *European Conference on Computer Vision*, pages 524–540. Springer, 2020. 1, 2, 3, 6, 7, 8

[29] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 1, 2, 3, 6, 7

[30] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, 2018. 2

[31] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015. 6

[32] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017. 2

[33] Samarth Sinha, Han Zhang, Anirudh Goyal, Yoshua Bengio, Hugo Larochelle, and Augustus Odena. Small-gan: Speeding up gan training using core-sets. In *International Conference on Machine Learning*, pages 9005–9015. PMLR, 2020. 2

[34] Rishabh Tiwari, Krishnateja Killamsetty, Rishabh Iyer, and Pradeep Shenoy. Gcr: Gradient coreset based replay buffer selection for continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 99–108, 2022. 1, 2

[35] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018. 2

[36] Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57, 1985. 2

[37] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *arXiv preprint arXiv:2302.00487*, 2023. 1

[38] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 374–382, 2019. 2, 6, 7

[39] Jaehong Yoon, Divyam Madaan, Eunho Yang, and Sung Ju Hwang. Online coreset selection for rehearsal-based continual learning. *arXiv preprint arXiv:2106.01085*, 2021. 1, 2, 3, 6, 7, 8

[40] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13208–13217, 2020. 2