Rethinking Momentum Knowledge Distillation in Online Continual Learning

Anonymous Authors[1]

Abstract

Online Continual Learning (OCL) addresses the problem of training neural networks on a continuous data stream where multiple classification tasks emerge in sequence. In contrast to offline Continual Learning, data can be seen only once in OCL, which is a very severe constraint. In this context, replay-based strategies have achieved impressive results and most state-of-the-art approaches heavily depend on them. While Knowledge Distillation (KD) has been extensively used in offline Continual Learning, it remains under-exploited in OCL, despite its high potential. In this paper, we theoretically analyze the challenges in applying KD to OCL. We introduce a direct yet effective methodology for applying Momentum Knowledge Distillation (MKD) to many flagship OCL methods and demonstrate its capabilities to enhance existing approaches. In addition to improving existing state-of-the-arts accuracy by more than 10% points on ImageNet100, we shed light on MKD internal mechanics and impacts during training in OCL. We argue that similar to replay, MKD should be considered a central component of OCL.

1. Introduction

Over the past decade, Deep Neural Networks (DNNs) have demonstrated super-human performance in most vision tasks (He et al., 2016; Redmon et al., 2016; Caron et al., 2021; Khosla et al., 2020). Nonetheless, current training procedures rely on strong assumptions. Specifically, during training, it is typically assumed that: 1) available data is independently and identically distributed (i.i.d.), and 2) all training data can be seen multiple times. Contrary to humans, DNNs are known to underperform or fail outright when these assumptions are not satisfied and suffer from

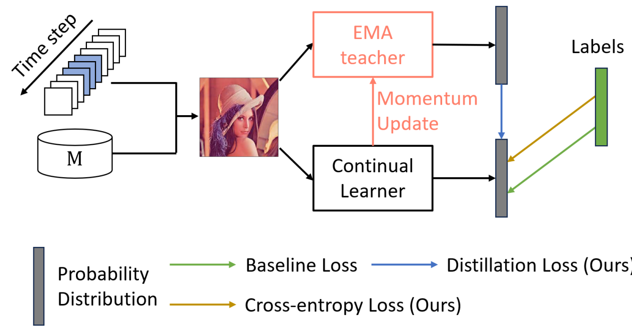


Figure 1. Overview of our MKD framework when applied to a baseline OCL method. Contrary to taking a snapshot at the end of each task, dynamic teacher address the key obstacles in OCL: teacher quality, teacher quantity, and unknown task boundaries.

Catastrophic Forgetting (CF) (French, 1999; Kirkpatrick et al., 2017). Addressing these challenges, Online Continual Learning (OCL) explores methods to mitigate CF in scenarios that violate assumptions 1) and 2). This is done by learning from a continuous stream of **non-i.i.d. data** where **only one pass is allowed**. Formally, OCL considers a sequential learning setup with a sequence $\{\mathcal{T}_1, \cdots, \mathcal{T}_K\}$ of K tasks, and $\mathcal{D}_k = (X_k, Y_k)$ the corresponding data-label pairs. For any value $k_1, k_2 \in \{1, \cdots, K\}$, if $k_1 \neq k_2$ then $Y_{k_1} \cap Y_{k_2} = \emptyset$. This scenario is known to be especially difficult and numerous approaches have been proposed to address it (He & Zhu, 2022; Guo et al., 2022; Mai et al., 2022; 2021; Caccia et al., 2022; Aljundi et al., 2019a; Guo et al., 2023; Prabhu et al., 2020; Aljundi et al., 2019b; Koh et al., 2023; Michel et al., 2023). In this study, we focus on the Class Incremental Learning scenario (Hsu et al., 2018) for OCL.

Among various methods, Experience Replay (ER) approaches (Rolnick et al., 2019; Buzzega et al., 2020; Khosla et al., 2020; Guo et al., 2022; Caccia et al., 2022; Michel et al., 2023; Guo et al., 2023) have demonstrated superior performances in OCL. The main component of this strategy is to store a small portion of previous samples to be used when training on new incoming samples. Current state-of-the-art methods in OCL mostly rely on combining replay strategies and specific loss designs. Unlike ER, only a few applications of Knowledge Distillation (KD) to OCL exist and present various limitations. DER (Buzzega

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

et al., 2020) stores previous sample logits and leverages knowledge distillation with ER but yields low performances. While MMKDDA (Han & Liu, 2022) tackles meta-learning with multi-level KD, it requires knowledge of total number of tasks and is computation intensive. Recently, SDP (Koh et al., 2023) proposes a hypo-exponential teacher for feature distillation in addition to ER. Even though SDP does not require task boundaries, it remains computationally expensive and architecture-dependent. In this work, we argue that KD has been rather overlooked by previous studies and can be efficiently adapted to OCL. Indeed, we believe that similarly to ER, KD plays an essential role in OCL and can be seamlessly combined with existing approaches.

Understanding the challenges specific to OCL is the key to explain why KD is not widely adopted in this context. Thus, we identify the three main KD challenges in OCL: Teacher Quality, Teacher Quantity and Unknown Task Boundaries. To overcome these challenges, we propose to take advantage of Momentum Knowledge Distillation (MKD) (Caron et al., 2021). Although MKD is a straightforward strategy, our technical contribution is a procedure which allows us to seamlessly integrate MKD with existing state-of-the-art approaches and show considerable improvements, even when compared to other distillation methods. Additionally, we highlight that utilizing MKD for OCL addresses prominent OCL challenges such as task-recency bias (Chrysakis & Moens, 2023; Mai et al., 2021), last layer bias (Liang et al., 2023; Ahn et al., 2021; Mai et al., 2021; Wu et al., 2019), feature drift (Caccia et al., 2022) and feature discrimination. In summary, the contributions of this paper are as follows:

- We identify the three main obstacles in applying KD to OCL and leverage MKD as a solution to overcome these challenges;
- We propose a strategy to seamlessly combine MKD with existing approaches and give insights on MKD internal mechanics and impacts during training in OCL;
- We experimentally demonstrate that MKD can significantly enhance the performance of existing methods.

2. Related Work

2.1. Knowledge Distillation in CL

We review KD strategies in both offline and online CL. We define offline CL as the multi-epoch CL training.

KD in offline CL Knowledge Distillation (KD) (Hinton et al., 2015) aims at transferring knowledge from a teacher model to a student model. This can be done by aligning their outputs, either in the logits space (Hinton et al., 2015; Romero et al., 2014; Zhao et al., 2022) or in the representation space (Aguilar et al., 2020; Tian et al., 2019). There are numerous KD applications in offline CL (Ahn et al., 2021; Douillard et al., 2020; Rebuffi et al., 2017; Cha et al., 2021; Simon et al., 2021; Hou et al., 2018; Wang et al., 2022). A common practice is to save the model at the end of each task, treating it as a snapshot, and use this model as a teacher for distillation during subsequent task trainings (Hou et al., 2018; Cha et al., 2021). Given that each teacher has task-specific knowledge, SS-IL (Ahn et al., 2021) leverages task-wise KD. There are also strategies that incorporate spatial distillation (Douillard et al., 2020) or feature compression (Wang et al., 2022).

KD in online CL Although KD has been widely adopted in offline CL, its adoption in OCL remains limited. DER (Buzzega et al., 2020) retains logits as well as data in memory for distillation in later stages. MMKDDA (Han & Liu, 2022) addresses meta-learning using multi-scale KD. Recently, SDP (Koh et al., 2023) introduced a teacher defined as a hypo-exponential moving average of current model for feature distillation. Nonetheless, these methods have their own constraints. DER exhibits suboptimal performance and scales poorly; MMKDDA requires task boundaries and is resource-intensive; SDP is architecture dependent and computationally expensive.

2.2. Blurry Task Boundaries

A common assumption in CL is that task boundaries are distinctly recognized during training. Similar to the work of (Michel et al., 2023), we refer to this as *clear* task boundaries. In OCL, however, we work on a continuous stream of incoming data, which makes *clear* boundaries unrealistic. In that sense, the concept of *blurry* task boundary setting has emerged in recent studies (Caccia et al., 2022; Michel et al., 2023; Bang et al., 2022). The idea is to have a gradual transition between tasks with an intermediate stage where data from both tasks are available in the stream. In this study, we embrace the perspective of unknown task boundaries, referring to it as the *blurry* setting, in opposition to the traditional *clear* setting as in (Michel et al., 2023).

2.3. Evaluation Metrics

We use the accuracy averaged across all tasks after training on the last task to compare the methods under consideration. This metric is commonly known as the final average accuracy (Kirkpatrick et al., 2017; Hsu et al., 2018). For highlighting the benefits of our approach for retaining past knowledge, we also take into account the Backward Transfer (BT) metric (Mai et al., 2022; Wang et al., 2023).

3. Challenges of KD in OCL

In this section, we discuss unique challenges in OCL that make implementation of KD in this context laborious.

Training Scenario	Accuracy (%)
Offline CL	81.8
Online CL	61.0
Online CL, Hard task	51.6
Online CL, Easy task	72.1

Table 1. Accuracy of GSA (Guo et al., 2023) on the first task of CIFAR100 M=5k splited in 10 tasks, on different training scenarios. We train for 20 epochs for Offline CL, 1 epoch for Online CL.

3.1. Teacher Quality

Given that incoming data can be seen by the model only once, it is uncertain whether the model has been fully trained at the end of each task. Consequently, taking a snapshot of the model at the end of the previous task may result in a suboptimal teacher. Such a teacher might hinder the student model's training for the subsequent task, leading to further degradation in the quality of teachers for the next task and an overall decline in performance. This problem is magnified when starting from a randomly initialized model, which is a common practice in OCL. Moreover, a model's performance on a specific task greatly depends on the difficulty of said task. Starting with a difficult task can lead to an especially low-quality teacher, further harming the distillation process.

Examples of such performance gaps are shown in Table 1 with GSA (Guo et al., 2023), a state-of-the-art approach. It can be observed that training offline leads to significantly higher performance than training online. Similarly, beginning training with an easy task induces superior performance on said task when compared with a hard task.

3.2. Teacher Quantity

One strategy for applying KD to CL requires taking a snapshot of the model at the end of each task (Rannen et al., 2017; Ahn et al., 2021; Hou et al., 2018). Each snapshot then serves as a teacher for the respective task and is incorporated into the distillation loss. Naturally, this requires storing a copy of the model per task which can be problematic for a large number of tasks, even in standard CL. We emphasize that memory consumption is crucial to OCL because it is presumed that only a small fraction of data can be retained, and all other incoming data is discarded post-usage. Dealing with a growing quantity of teachers is unrealistic and contradicts the implicit storage constraint of the online setup.

To circumvent the issue of continuously increasing teacher numbers, one might consider using just the snapshot from the most recent task as a teacher. However, this solution is also unsatisfactory as this teacher should encapsulate the knowledge from all previous tasks, which is especially complex for long task sequences.

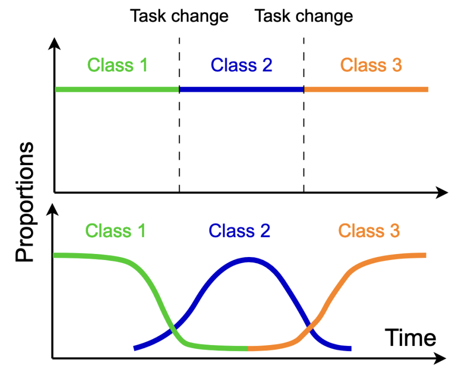


Figure 2. Illustration of the *blurry* boundary setting (bottom row) in opposition to the *clear* boundary setting (top row). Detecting task change in the case of *blurry* is not trivial.

3.3. Unknown Task Boundaries

Most distillation strategies in CL rely on task boundaries information to select the best teachers for distillation. In offline CL, this information is easily available. However in OCL, pinpointing the exact moment of task change is not guaranteed. Figure 2 illustrates a more realistic scenario where transitions occur progressively, making the determination of the ideal snapshot moment challenging. Choosing a suboptimal teacher can also compromise the quality of distillation.

4. Methodology

4.1. Motivations

As mentioned in previous sections, KD has been underutilized in OCL. The main reason is that most KD strategies draw inspiration from offline CL where the teacher is typically frozen at the conclusion of the previous task. However, relying on a frozen teacher in OCL can be problematic due to unknown task boundaries and concerns regarding teacher quality. Moreover, a static teacher from the previous task will set an upper limit on the student's learning potential. Consequently, the student is unable to enhance performance on the previous task while mastering the current one. In other words, a simple teacher discourages backward transfer.

To tackle this limitation, we propose the use of an evolving teacher. Contrary to a fixed teacher, the weights of an evolving teacher are updated throughout the training process. This approach allows the teacher to continually improve and not hinder the student's progression. A student learning from an evolving teacher can consistently refine their performance on preceding tasks, thereby promoting backward transfer. Additionally, this kind of teacher eliminates the need for the knowledge of task boundaries. In this paper, we take

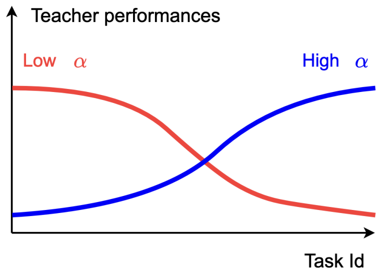


Figure 3. Impact of α on the plasticity-stability trade-off. Lower α values imply a stable teacher with high performances on old tasks. Higher α implies a plastic teacher, with high performances on new tasks.

advantage of an Exponential Moving Average (EMA) of the current model as the evolving teacher and design a novel MKD teacher-dependent weighting scheme for adapting MKD to OCL. While EMA can efficiently solve previously described challenges, its applications to OCL is still in its infancy.

4.2. Momentum Knowledge Distillation

We propose a new scheme to leverage Momentum Knowledge Distillation (MKD) with an evolving teacher. In this distillation strategy, the teacher architecture mirrors that of the student and its weights are computed as an Exponential Moving Average of the student parameters. The EMA weights are computed online according to the update parameters α such that:

$$\theta_\alpha(t) = \alpha * \theta(t) + (1-\alpha) * \theta_\alpha(t-1), \tag{1}$$

where $\theta(t)$ represents the student's model parameters at time t. The teacher, parameterized by θ_α, is represented as $\mathcal{T}_\alpha$.

4.3. Rethinking MKD

Plasticity-stability control When designing CL methods, it is common to address the plasticity-stability tradeoff (Wang et al., 2023). Usually, the application of distillation augments the model's stability at the expense of its plasticity. Using Momentum Knowledge Distillation gives a precise control over this trade-off through the parameter α. A lower value of α would make the teacher update slower and remember longer timelines, making it retain longer timelines but offering scant knowledge on the current task. A high value of α would help the student learn the current task but with limited insight of previous tasks. In other words, a higher value of α emphasizes plasticity over stability whereas a lower value of α encourages stability over plasticity. This plasticity-stability control characteristic is illustrated in Figure 3. We make concrete usage of this property by designing a teacher-dependent weighting scheme in our model learning.

Algorithm 1 PyTorch-like pseudo-code of our loss to integrate to other baselines.

```
for x, y in dataloader:
  # Baseline loss
  loss_baseline = criterion_baseline(model, x, y)
  loss = loss_baseline

  # Proposed loss
  x_aug = transform(x)  # data augmentation
  l_stu1 = model(x) # logits student
  l_stu2 = model(x_aug) # logits student
  l_tea = teacher(x_aug) # logits teacher
  loss_ce = cross_entropy(x_aug, y)
  loss_d1 = kl_div(softmax(l_stu1/t), softmax(l_tea/t)) #
↪   temperature t
  loss_d2 = kl_div(softmax(l_stu2/t), softmax(l_tea/t))
  loss_dist = (loss_d1 + loss_d2)/2  # Eq. 3
  loss += loss_ce + lam*loss_dist

  optim.zero_grad()
  loss.backward()
  optim.step()
  update_ema()
```

Model learning We formulate our loss term using an EMA teacher as described in equation 2.

$$\begin{aligned}\mathcal{L}(X,Y) =& \mathcal{L}_{CE}(X,Y) + \\ & \lambda_\alpha * KL(\mathcal{T}_\alpha(X)/\tau, S(X)/\tau),\end{aligned} \tag{2}$$

where $\mathcal{L}_{CE}$ the Cross-Entropy function, λ_α a weighting hyper-parameter depending on α, S the student model, (X, Y) the data-label pairs, KL the Kullback–Leibler divergence and τ the distillation temperature. We further introduce multiview distillation, by making use of a data augmentation procedure $Aug(.)$ and propose to minimize $\mathcal{L}_{MKD}$ defined in Equation 3.

$$\begin{aligned}\mathcal{L}_{MKD}(X,Y) =& \mathcal{L}_{CE}(\hat{X},Y) \\ & + \frac{\lambda_\alpha}{2} KL(\mathcal{T}_\alpha(X), S(\hat{X})) \\ & + \frac{\lambda_\alpha}{2} KL(\mathcal{T}_\alpha(\hat{X}), S(\hat{X})),\end{aligned} \tag{3}$$

where $\hat{X} = Aug(X)$.

The only hyper-parameter is α. In Section 6, we give details on how to efficiently choose α and how to express the teacher-dependent weighting parameter λ_α. Additionally, the simplicity of this process allows for seamless adaptation to existing methods. We provide a PyTorch-like (Paszke et al., 2019) pseudo-code that outlines the strategy for integrating our proposed MKD into other training procedures, as can be found in Algorithm 1.

In this pseudo-code, we have omitted a memory buffer for simplicity. Nonetheless, the training procedure remains consistent, using a batch combining stream and memory data.

Model estimation As introduced in the plasticity-stability control section, the knowledge of the teacher and student pertains to different tasks. The student is inclined towards the current task whereas the teacher excels in past tasks. Solely relying on the teacher's or student's weights for inference may not yield optimal performances. Consequently, we introduce a new model estimation strategy that necessitates minimal extra computation. We compute the final model parameters $\theta^\star$ as the average of teacher and student weights such that $\theta^\star = \frac{\theta_S+\theta_T}{2}$, where θ_S and θ_T denote the parameters of the student and teacher, respectively. We show in Section 5.4 that this strategy can enhance performance.

5. Experiments

5.1. Implementation Details

For each method, we use random retrieval and reservoir sampling (Vitter, 1985) for memory management. We use a full ResNet18 (He et al., 2016) (untrained) for every method. For all baselines, we perform a small hyperparameter search on CIFAR100, M=5k, applying the determined parameters across other configurations. More details are given in the Appendix. We use the same hyperparameters when incorporating our loss. Throughout the training process, the streaming batch size is set to 10, and data retrieval from memory is capped at 64. Data augmentation includes random flip, grayscale, color jitter, and random crop. The *blurry* datasets are created following the code given in (Michel et al., 2023) with a scale of 500. Some methods require task boundary inference to be adapted to the *blurry* setting, which is detailed in Appendix. The temperature τ designated for KD is 4. For MKD, we use $\alpha = 0.01$ and $\lambda_\alpha = 5.5$ accordingly for every method. For more details regarding experiments, please refer to the Appendix.

5.2. Baselines

To show the efficiency of our proposed approach, we integrate our approach as described in our pseudo code into several baselines and the state-of-the-art methods in OCL. **ER** (Rolnick et al., 2019): A basic memory based method leveraging a Cross-Entropy loss and a replay buffer. **DER++** (Buzzega et al., 2020): A replay-based approach doing distillation of old stored logits with using task boundaries. **ER-ACE** (Caccia et al., 2022): A replay-based method using an Asymmetric Cross Entropy to overcome feature drift. **DVC** (Gu et al., 2022): A replay-based approach leveraging consistency between image views in addition to minimizing cross entropy. **OCM** (Guo et al., 2022): A replay-based method maximizing mutual information between old and new samples representation. **GSA** (Guo et al., 2023): A replay-based method dealing with cross-task class discrimination with a redefined loss objective using Gradient Self Adaptation. **PCR** (Lin et al., 2023): A replay-based method leveraging a proxy-based contrastive loss for OCL. For the sake of reproducibility, we re-implemented the methods mentioned above and will make the code public upon acceptance.

5.3. Experimental Results

Clear boundary setting To demonstrate the effectiveness of our approach, we applied the procedure described to all the considered baselines and compared the performances. Average accuracy at the end of training for the *clear* setting is displayed in Table 2. It can be observed that for most of the considered methods, datasets and memory sizes, applying our procedure improves performance. In most cases, this gain in performance is significant. Specifically, the combinations *GSA + ours* and *OCM + ours* have the potential to surpass the current state-of-the-art methods. Additionally, the standard deviation is also significantly reduced when applying our approach, showing that the use of a momentum teacher can help stabilizing the training procedure. More interestingly, the introduction of our distillation procedure can enhance performance, even if distillation is already incorporated in the method (e.g., *DER++*).

Blurry boundary setting To further demonstrate the capabilities of MKD, we also conducted experiments with *blurry* task boundaries. Average accuracy at the end of training is shown in Table 2. However, we did not implement GSA in this context since it requires knowledge of the exact class-task relationships and is not easily adaptable to this setup. Additionally, we inferred task boundaries for OCM, since it is required to apply the method. Details on how the task boundaries are inferred in this setup are given in Appendix. Similar to the *clear* boundary setting, incorporating MKD as per our procedure can significantly enhance performance. This performance gain becomes even more pronounced when the original method experiences a drop in effectiveness due to the challenging nature of the setting. For example, OCM performances on CIFAR100 M=5k drop from 41.87% to 38.14% while *OCM + ours* performances remain stable around 51.4%.

Comparison with SDP SDP (Koh et al., 2023) uses a hypo-exponential evolving teacher, akin to our approach. While initially proposed as a standalone method, SDP can be combined with existing techniques. We integrated SDP with *ER* and *GSA*, and results in Table 2 reveal that, although SDP enhances *ER*, *ER + SDP* performs less effectively than *ER + ours*. Additionally, for *GSA*, the inclusion of SDP leads to decreased performance, confirming MKD's superiority over SDP. Computationally, as SDP operates in representation space, it demands more resources compared to MKD, which is computed in logit space. Further details on the computational constraints are provided in the Appendix. The introduction of SDP has a more substantial impact on the time consumption of *ER* and *GSA* than MKD.

Dataset	CIFAR10			CIFAR100			Tiny-IN			ImageNet100		
Memory Size M	200	500	1000	1000	2000	5000	2000	5000	10000	2000	5000	10000
ER [NeurIPS'19]	46.33±2.42	55.73±2.04	62.99±2.1	23.0±0.8	31.55±1.27	38.05±1.08	11.39±0.75	18.97±1.16	21.52±3.37	19.06±0.9	29.74±1.34	36.72±1.09
ER + SDP	47.78±2.29	58.85±1.38	65.93±2.15	28.48±1.18	35.21±0.73	41.19±1.11	15.32±0.47	23.22±0.31	26.97±1.1	22.95±0.13	32.12±0.23	35.64±0.25
ER + ours	**57.54±2.55**	**68.48±0.92**	**74.33±0.68**	**38.5±0.5**	**45.2±0.2**	**52.10±0.5**	**23.95±0.65**	**32.22±0.88**	**38.27±0.18**	**30.67±0.46**	**39.7±1.07**	**44.92±0.98**
DER++ [NeurIPS'20]	47.07±0.97	55.53±1.05	58.51±0.68	22.8±1.8	25.89±1.46	25.71±2.4	3.89±0.64	4.28±0.51	4.16±0.32	15.36±3.04	19.19±1.55	20.48±4.67
DER++ + ours	**53.63±2.18**	**63.95±0.86**	**68.84±1.15**	**32.1±0.5**	**37.97±0.92**	**41.97±1.53**	**17.08±1.43**	**15.64±4.64**	**13.69±3.36**	**26.18±1.07**	**33.85±0.98**	**38.22±1.84**
ERACE [ICLR'22]	44.77±3.18	52.65±1.37	61.45±1.47	27.4±0.6	32.88±0.63	39.61±0.53	14.79±0.95	22.25±1.69	26.64±0.91	27.16±0.57	32.88±0.83	39.14±0.35
ERACE + ours	**58.99±1.36**	**65.94±0.49**	**69.78±0.96**	**37.0±0.7**	**42.92±0.79**	**48.73±1.29**	**22.21±0.87**	**31.13±0.41**	**35.54±0.43**	**33.59±0.99**	**41.93±0.64**	**47.16±0.89**
DVC [CVPR'22]	48.08±4.27	58.72±2.03	61.11±2.97	18.66±2.54	22.73±2.9	28.47±3.95	2.04±0.8	1.47±0.49	1.54±0.79	14.54±5.15	21.88±3.45	28.5±2.93
DVC + ours	**50.53±4.35**	**62.62±1.84**	**69.52±0.84**	**27.42±3.14**	**35.95±1.71**	**42.45±2.45**	**9.41±1.43**	**12.03±3.83**	**13.44±3.84**	**18.75±1.96**	**29.64±3.6**	**38.0±2.94**
OCM [ICML'22]	59.58±1.43	68.46±0.79	72.79±2.36	29.3±1.55	36.7±0.58	41.87±1.52	19.58±0.63	27.85±1.03	32.56±1.37	28.7±0.92	37.37±1.11	41.86±1.14
OCM + ours	**67.02±3.14**	**75.14±0.75**	**79.33±0.55**	**38.21±0.62**	**45.51±0.94**	**51.24±0.81**	**23.07±0.37**	**31.82±0.72**	**37.46±0.95**	**28.87±1.85**	**38.26±1.06**	**44.24±0.55**
GSA [CVPR'23]	48.9±3.38	61.45±1.95	67.63±1.24	29.68±1.54	36.96±0.79	45.86±1.89	15.77±0.72	22.48±0.4	28.46±1.85	24.29±0.59	33.47±1.18	40.18±0.93
GSA + SDP	47.39±1.76	60.61±3.43	67.17±1.41	26.56±3.03	34.78±3.72	44.53±1.07	11.71±2.69	16.1±6.3	25.92±3.05	27.7±10.69	43.85±1.42	51.39±1.07
GSA + ours	**57.66±4.11**	**68.16±0.85**	**75.08±1.14**	**40.1±1.0**	**48.23±0.78**	**56.15±0.6**	**23.14±0.44**	**32.38±1.28**	**38.78±0.65**	**33.4±0.85**	**44.99±0.46**	**52.41±0.59**
PCR [CVPR'23]	52.2±0.66	60.61±2.23	61.66±13.86	30.68±0.81	38.63±1.01	45.27±0.78	12.47±3.56	20.41±2.84	23.85±4.21	19.89±6.24	31.35±3.01	36.99±4.7
PCR+ours	**55.83±2.35**	**67.03±1.33**	**73.47±0.53**	**35.27±0.47**	**44.95±0.44**	**54.44±0.46**	**17.14±0.48**	**29.05±0.55**	**36.65±0.90**	**25.42±0.54**	**39.50±1.00**	**49.66±1.78**

Table 2. Final average accuracy (%) for the *clear* boundary setting at the end of training for considered baselines, with and without our additional MKD procedure. Results are displayed for different datasets and memory sizes. Displayed values are the mean and standard deviation computed over 5 runs for ImageNet100 and 10 runs for other datasets.

Dataset	CIFAR10			CIFAR100			Tiny-IN			ImageNet100		
Memory Size M	200	500	1000	1000	2000	5000	2000	5000	10000	2000	5000	10000
ER [NeurIPS'19]	44.78±6.22	54.1±4.54	64.17±1.89	24.76±1.33	31.56±1.73	39.1±1.0	11.88±1.5	19.76±1.67	25.71±1.29	15.18±1.51	24.88±1.27	31.44±2.18
ER + SDP	47.64±1.66	60.04±1.53	65.89±0.84	28.66±1.61	36.07±1.5	41.65±1.47	15.98±1.6	23.91±1.48	28.92±0.84	14.26±1.47	5.14±3.14	5.15±2.16
ER + ours	**56.69±1.9**	**69.15±1.37**	**74.06±1.01**	**38.38±0.86**	**45.47±0.63**	**51.79±0.22**	**25.08±0.64**	**33.22±0.64**	**38.63±0.8**	**26.03±0.76**	**36.67±0.65**	**42.64±0.87**
DER++ [NeurIPS'20]	47.28±2.03	55.83±2.45	59.37±1.93	23.4±1.54	27.91±1.3	29.31±1.83	15.99±0.9	20.34±1.22	21.36±0.81	3.65±1.38	3.98±1.53	4.22±1.66
DER++ + ours	**54.21±3.11**	**63.83±1.83**	**69.06±1.7**	**31.17±0.81**	**38.44±1.15**	**42.72±0.81**	**21.93±0.74**	**28.7±0.55**	**32.58±1.51**	**20.69±1.09**	**27.37±0.93**	**30.25±1.01**
ERACE [ICLR'22]	50.44±1.37	56.5±1.77	62.92±1.5	27.69±1.45	32.98±0.81	40.12±1.05	19.04±0.88	25.27±1.27	30.05±1.67	15.84±0.82	24.49±0.3	31.74±0.97
ERACE + ours	**59.36±3.15**	**66.25±1.82**	**70.74±0.8**	**38.04±0.93**	**43.75±0.46**	**50.35±0.48**	**25.85±0.7**	**33.14±0.79**	**37.68±0.57**	**26.14±0.64**	**35.61±1.06**	**42.67±0.81**
DVC [CVPR'22]	46.05±5.23	58.73±2.4	58.78±5.83	22.46±1.91	26.98±3.13	29.46±2.39	10.64±1.31	15.48±2.1	15.81±1.76	2.97±0.65	5.34±2.49	8.38±4.58
DVC + ours	**49.04±2.95**	**61.95±1.81**	**69.25±0.78**	**27.1±2.11**	**35.76±2.27**	**41.99±3.31**	**12.45±2.19**	**22.15±1.42**	**24.14±3.63**	**6.53±1.12**	**15.32±1.28**	**5.42±1.35**
OCM [ICML'22]	43.66±2.59	47.63±2.68	51.08±2.66	25.16±0.76	32.96±1.21	38.14±1.11	18.57±0.37	26.82±0.86	31.21±0.55	**26.61±1.02**	**36.36±0.48**	**41.92±0.9**
OCM + ours	**67.66±0.49**	**74.9±0.98**	**78.61±0.43**	**36.64±0.47**	**44.63±1.12**	**51.41±0.71**	**24.77±0.2**	**33.01±1.1**	**39.39±0.89**	25.52±1.27	34.42±0.97	39.07±0.8
PCR [CVPR'23]	53.43±2.2	60.67±3.29	69.13±0.66	30.9±2.06	38.63±0.26	45.97±1.18	16.0±2.35	22.02±3.21	28.9±3.73	9.77±4.75	16.55±7.91	27.86±5.46
PCR+ours	**57.55±1.4**	**67.03±2.0**	**74.0±0.91**	**35.6±0.66**	**44.95±0.42**	**54.87±0.39**	**17.33±1.28**	**29.58±0.6**	**38.02±1.64**	**22.51±0.96**	**34.53±0.57**	**44.28±0.68**

Table 3. Final average accuracy (%) for the *blurry* boundary setting at the end of training for considered baselines, with and without our additional MKD procedure. Results are displayed for different datasets and memory sizes. Displayed values are the mean and standard deviation computed over 5 runs for ImageNet100 and 5 runs for other datasets.

5.4. Ablation Studies

Impact of the final weight estimation To demonstrate the impact of averaging weights from the teacher and the student, we experimented using either the teacher or the student exclusively for inference. Results are displayed for *ER* on Table 4. In both cases, employing solely the student or the teacher results in inferior performance compared to using their averaged weights, with a minimum drop in accuracy of 0.5%. Additionally, the teacher performs worse than the student, which can be due to the fact that for remembering enough from past tasks, the teacher update must be quite slow. In that sense, the teacher might perform worse overall but improve the students' stability.

Impact of multiview distillation As described in the Model Learning section, we employ both augmented and raw images (two views) in our distillation process. In Table 4 we show the performance of *ER + ours* when trained using only one view. Namely, minimizing $\mathcal{L}(X, Y) = \mathcal{L}_{CE}(\hat{X}, Y) + \lambda_\alpha KL(\mathcal{T}_\alpha(\hat{X}), S(\hat{X}))$. The results indicate that employing this multiview distillation strategy has a significant impact, yielding at least a 2.9% points boost in accuracy.

Dataset	CIFAR100		
Memory Size M	1000	2000	5000
ER + ours	38.5±0.5	45.2±0.2	52.1±0.5
ER + ours (student)	37.7±0.7	44.7±0.5	51.2±0.6
ER + ours (teacher)	37.2±0.7	43.0±0.8	49.4±0.6
ER + ours (student, one view)	34.8±0.6	41.8±0.6	47.9±0.4

Table 4. Final average accuracy (%) on CIFAR100, *clear* boundary setting, for *ER + ours* and varying memory sizes. *Student* corresponds to the student performance and *teacher* to the teacher performance. *no aug* corresponds using the distillation loss with only one view as defined in Section 5.4. Mean and standard deviations over 5 runs are displayed.

6. Discussions

In this section, we analyze the working mechanisms of MKD for OCL.

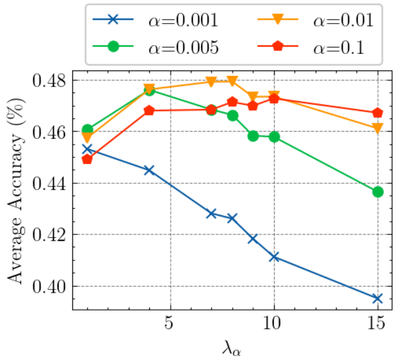


Figure 4. Impact of λ_α and α on the final performances or *ER* on CIFAR100 M=5k, *clear* setting.

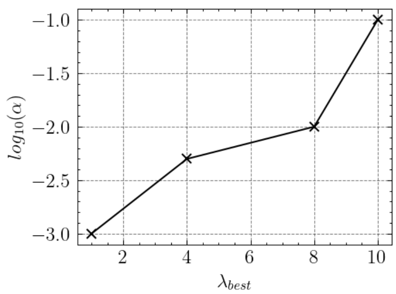


Figure 5. Relation between $\log \alpha$ and and the best corresponding λ_α value, λ_{best}. The displayed relation is linear.

6.1. Choosing α

Since α directly influences the teachers' knowledge, it has a significant impact on performances. Finding the best value of α can be done by grid search. Figure 4 shows the final average accuracy for various values of (α, λ_α), in log scale for *ER + Ours* on CIFAR100 M=5K. To avoid computation-intensive grid search, we show in the subsequent section that α can be selected from a broad range, provided the relation between α and λ_α is maintained.

6.2. Expressing λ_α

Figure 5 illustrates a strong interdependence between α and λ_α. The optimal value for λ_α given α follows the formula $\lambda_\alpha = a * \log_{10}(\alpha) + b$, with $a = 9/2$ and $b = 29/2$. Notably, lower values of α correspond to lower values of λ. This correlation arises from the fact that a larger α leads to a teacher closely resembling the student, resulting in a low distillation loss and a higher λ_α for compensation.

6.3. Reducing Task-Recency Bias

A common issue in Continual Learning is the task-recency bias (Chrysakis & Moens, 2023; Mai et al., 2021). This is the problem of over-predicting the classes belonging to the last task seen. Figure 8 displays confusion matrices at the end of training for considered baselines, with and without MKD. While most baselines suffer from task-recency bias at the end of training, it can be observed qualitatively that adding MKD reduces this bias by diminishing the amount of last task false positives.

Method	Logits Acc.	NCM Acc.
ER [NeurIPS'19]	23.0±0.8	29.0±0.3 (↑6.0)
ER + ours	38.5±0.5	31.5±0.5 (↓7.0)
DER++ [NeurIPS'20]	22.8±1.8	26.6±3.4 (↑3.8)
DER++ + ours	32.1±0.5	28.7±1.4 (↓3.4)
ERACE [ICLR'22]	27.4±0.6	28.1±0.7 (↑0.7)
ERACE + ours	37.0±0.7	34.2±0.2 (↓2.8)
GSA [CVPR'23]	29.7±1.5	32.6±1.6 (↑2.9)
GSA + ours	40.1±1.0	36.2±0.5 (↓3.9)

Table 5. Final Average Accuracy (%) on CIFAR100 M=1k of several baselines, with and without using the NCM trick. Logits Acc. refers to the accuracy of the model using predicted logits while NCM Acc. refers to NCM accuracy trained on intermediate representations from memory at the end of training.

6.4. Reducing Last Layer Bias

Another identified issue when training with Cross Entropy is the presence of bias in the last Fully Connected (FC) layer (Liang et al., 2023; Ahn et al., 2021; Mai et al., 2021; Wu et al., 2019). To demonstrate the presence of the last FC bias, one can make use of the Nearest Class Mean (NCM) trick (Mai et al., 2021) with intermediate representations given by the model. Since we work with memory based approaches, we compare the model's performance using logits with performances obtained by training an NCM classifier using intermediate representation of memory data at the end of training. In other words, we drop the last FC layer and fine-tune with a simple NCM classifier on memory. The NCM trick yields substantial performance improvement in the presence of a pronounced last layer bias, as indicated in Table 5. Across various baselines, with and without MKD, the NCM trick consistently enhances performances, underscoring the influence of a strong last FC bias. Intriguingly, when our approach is applied to these baselines, leveraging the NCM actually leads to performance degradation. This suggests a neutralization of the last FC layer bias, possibly due to the distillation loss occurring in the logit space, where the last FC layer is tightly constrained.

6.5. Reducing Feature Drift

When training in OCL, one potential issue is the feature drift (Caccia et al., 2022). Feature drift occurs when changing tasks causes the representation of old classes to conflict with the representations of new classes, inducing large changes in past representations. Experimentally, we demonstrate that MKD can inherently reduce feature drift. Figure 6 shows the feature drift $d_t = ||f_{\theta_t}(X_{old}) - f_{\theta_{t+1}}(X_{old})||_2$, where X_{old} are memory images of *old* classes and f_{θ_t} is the model parameterized by θ from which we removed the last FC layer. As we can see, using MKD greatly reduces feature drift throughout training. For *ER + ours* (MKD), the

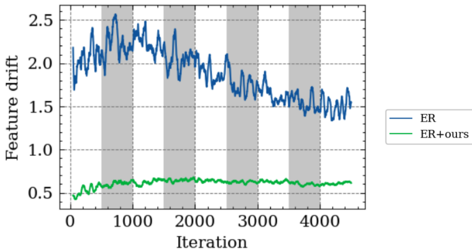
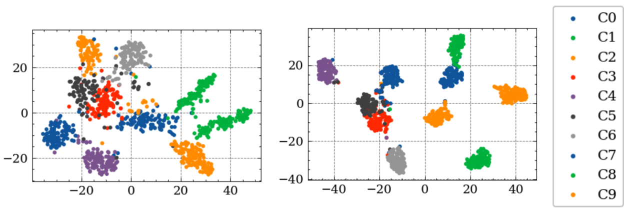


Figure 6. Feature drift d_t of *ER* and *ER + ours* (MKD) on CIFAR100 M=5k.



(a) ER (b) ER + ours (MKD)

Figure 7. (a) t-SNE of memory data at the end of training *ER* on CIFAR10, M=1k. (b) t-SNE of memory data at the end of training *ER + ours* (MKD) on CIFAR10, M=1k.

feature drift is not only lower but also more stable.

6.6. Improving Feature Discrimination

Feature discrimination is a desirable property of any learning process. Specifically in Continual Learning, it is important to obtain distinctive features at the end of training. In Figure 7, we present the t-SNE results on memory data at the end of training of *ER* and *ER + ours* (MKD). Clearly, the obtained representation using MKD is significantly more discriminative than the one obtained without MKD. Even though our distillation loss is proposed in the logit space, it can still greatly improve learned feature quality.

6.7. Improving Backward Transfer

As the plasticity-stability dilemma is central in Continual Learning, a variety of metrics have been designed to adequately measure either plasticity or stability (Mai et al., 2022; Wang et al., 2023). Knowledge Distillation is particularly interesting for remembering past information and enhancing the model's stability during training. To showcase this effect, we look at the Backward Transfer of considered baselines, with and without MKD. Table 6 shows the BT at the end of training. In every scenario, our method improves BT. Specifically, for *ER*, leveraging MKD can yield a positive backward transfer, implying that the models keep improving on old classes even after a task change. This property is especially important in OCL since the student is unlikely to have fully learned the past task when training on the current task.

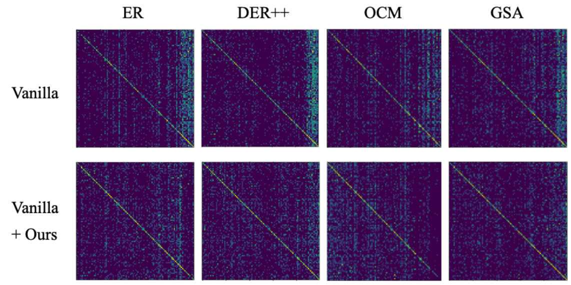


Figure 8. Confusion matrix on the evaluation set at the end of training on CIFAR100 with M=1K for considered baselines. Classes are shown in training order. The top row is the confusion matrices for baselines without the MKD procedure. The bottom row is the confusion matrices when adding MKD.

Method	CIFAR100	ImageNet100
ER	-16.7$_{\pm 1.2}$	-17.5$_{\pm 1.5}$
ER + ours	**+8.15$_{\pm 0.8}$**	**-1.3$_{\pm 2.3}$**
DER++	-27.5$_{\pm 3.4}$	-18.9$_{\pm 2.5}$
DER++ + ours	**-10.4$_{\pm 5.6}$**	**-14.4$_{\pm 2.5}$**
GSA	-4.9$_{\pm 1.2}$	-17.0$_{\pm 1.3}$
GSA + ours	**-2.5$_{\pm 3.1}$**	**-15.5$_{\pm 1.0}$**

Table 6. Backward Transfer (%) at the end of training on CIFAR100, M=5k and Imagenet100, M=10k for several baselines. Higher is better. Means over 5 runs are displayed.

7. Conclusions

In this paper, we studied the problem of Online Continual Learning from the perspective of Knowledge Distillation. While KD has been widely studied in the context of offline continual learning, it remains under-used in OCL. To understand the current state of KD in OCL, we identified OCL-specific challenges for applying KD: Teacher Quality, Teacher Quantity, and Unknown Task Boundaries. Moreover, we proposed to address these challenges by designing a new distillation procedure based on Momentum Knowledge Distillation. This approach benefits from a powerful plasticity-stability control for OCL and employs an evolving teacher to overcome the previously introduced challenges. We experimentally demonstrated the efficiency of our approach and achieved more than 10% points improvement over state-of-the-art methods on several datasets. Additionally, we provided insightful explanations on how using MKD can help solve multiple OCL known issues: task-recency bias, last layer bias, feature drift, feature discrimination, and backward transfer. Our approach is architecture-independent and computationally efficient. In conclusion, we have shed new light on distillation for OCL and advocate for its efficiency and its potential as a central component for addressing OCL.

8. Broader Impacts

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

References

Aguilar, G., Ling, Y., Zhang, Y., Yao, B., Fan, X., and Guo, C. Knowledge distillation from internal representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7350–7357, 2020.

Ahn, H., Kwak, J., Lim, S., Bang, H., Kim, H., and Moon, T. SS-IL: Separated Softmax for Incremental Learning. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Aljundi, R., Belilovsky, E., Tuytelaars, T., Charlin, L., Caccia, M., Lin, M., and Page-Caccia, L. Online Continual Learning with Maximal Interfered Retrieval. In *Advances in Neural Information Processing Systems*, volume 32, 2019a.

Aljundi, R., Lin, M., Goujaud, B., and Bengio, Y. Gradient based sample selection for online continual learning. *Advances in Neural Information Processing Systems*, 32, 2019b.

Bang, J., Koh, H., Park, S., Song, H., Ha, J.-W., and Choi, J. Online continual learning on a contaminated data stream with blurry task boundaries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9275–9284, 2022.

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

Caccia, L., Aljundi, R., Asadi, N., Tuytelaars, T., Pineau, J., and Belilovsky, E. New insights on reducing abrupt representation change in online continual learning, 2022.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, 2021.

Cha, H., Lee, J., and Shin, J. Co2l: Contrastive continual learning. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9516–9525, 2021.

Chrysakis, A. and Moens, M.-F. Online bias correction for task-free continual learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Douillard, A., Cord, M., Ollion, C., Robert, T., and Valle, E. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *16th European Conference on Conputer Vision (ECCV)*, pp. 86–102, 2020.

French, R. M. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

Gu, Y., Yang, X., Wei, K., and Deng, C. Not Just Selection, but Exploration: Online Class-Incremental Continual Learning via Dual View Consistency. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7432–7441, June 2022. doi: 10.1109/CVPR52688.2022.00729.

Guo, Y., Liu, B., and Zhao, D. Online Continual Learning through Mutual Information Maximization. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

Guo, Y., Liu, B., and Zhao, D. Dealing with cross-task class discrimination in online continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Han, Y.-n. and Liu, J.-w. Online Continual Learning via the Meta-learning update with Multi-scale Knowledge Distillation and Data Augmentation. *Engineering Applications of Artificial Intelligence*, 113, August 2022. ISSN 0952-1976.

He, J. and Zhu, F. Online continual learning via candidates voting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3154–3163, 2022.

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

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

Hou, S., Pan, X., Loy, C. C., Wang, Z., and Lin, D. Lifelong learning via progressive distillation and retrospection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 437–452, 2018.

Hsu, Y.-C., Liu, Y.-C., Ramasamy, A., and Kira, Z. Re-evaluating continual learning scenarios: A categorization and case for strong baselines. *arXiv preprint arXiv:1810.12488*, 2018.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

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

Koh, H., Seo, M., Bang, J., Song, H., Hong, D., Park, S., Ha, J.-W., and Choi, J. Online boundary-free continual learning by scheduled data prior. In *International Conference on Learning Representations*, 2023.

Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. 2009.

Le, Y. and Yang, X. S. Tiny ImageNet Visual Recognition Challenge. 2015.

Liang, G., Chen, Z., Chen, Z., Ji, S., and Zhang, Y. New Insights on Relieving Task-Recency Bias for Online Class Incremental Learning. February 2023.

Lin, H., Zhang, B., Feng, S., Li, X., and Ye, Y. Pcr: Proxy-based contrastive replay for online class-incremental continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24246–24255, 2023.

Mai, Z., Li, R., Kim, H., and Sanner, S. Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3589–3599, 2021.

Mai, Z., Li, R., Jeong, J., Quispe, D., Kim, H., and Sanner, S. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469:28–51, 2022.

Michel, N., Chierchia, G., Negrel, R., and Bercher, J.-F. Learning Representations on the Unit Sphere: Application to Online Continual Learning. *arXiv preprint arXiv:2306.03364*, June 2023. doi: 10.48550/arXiv.2306.03364.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

Prabhu, A., Torr, P. H., and Dokania, P. K. Gdumb: A simple approach that questions our progress in continual learning. In *Computer Vision–ECCV 2020: 16th European Conference, Proceedings, Part II 16*, pp. 524–540, 2020.

Rannen, A., Aljundi, R., Blaschko, M. B., and Tuytelaars, T. Encoder based lifelong learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1320–1328, 2017.

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

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, 2016.

Rolnick, D., Ahuja, A., Schwarz, J., Lillicrap, T., and Wayne, G. Experience Replay for Continual Learning. In *Advances in Neural Information Processing Systems*, 2019.

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

Simon, C., Koniusz, P., and Harandi, M. On learning the geodesic path for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1591–1600, 2021.

Tian, Y., Krishnan, D., and Isola, P. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019.

Vitter, J. S. Random sampling with a reservoir. *ACM Transactions on Mathematical Software*, 11(1):37–57, March 1985. ISSN 0098-3500, 1557-7295.

Wang, F.-Y., Zhou, D.-W., Ye, H.-J., and Zhan, D.-C. Foster: Feature boosting and compression for class-incremental learning. In *European Conference on Computer Vision*, pp. 398–414, 2022.

Wang, L., Zhang, X., Su, H., and Zhu, J. A Comprehensive Survey of Continual Learning: Theory, Method and Application. *arXiv preprint arXiv:2302.00487*, January 2023.

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

Zhao, B., Cui, Q., Song, R., Qiu, Y., and Liang, J. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11953–11962, 2022.

A. Additional Experiments

A.1. Task-Recency Bias

In the main paper, we discussed how our approach addresses the task-recency bias in OCL for only a limited number of methods due to space constraints. In Figure 9, we share confusion matrices for every considered method from the main paper.

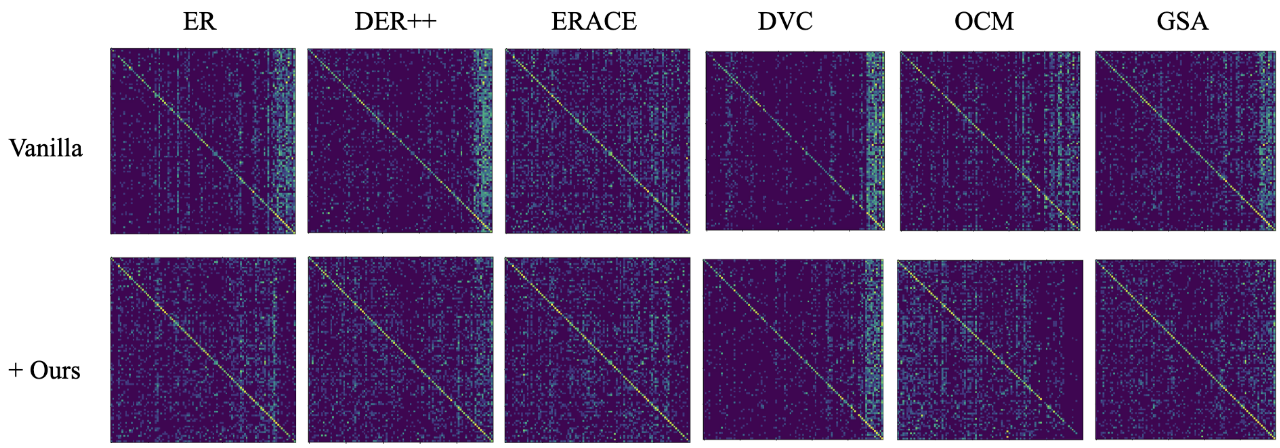


Figure 9. Confusion matrix on the evaluation set at the end of training on CIFAR100 with M=1K for considered baselines. Classes are shown in the same order of those during training such that left columns of confusion matrices correspond to first classes seen during training. The top row presents the confusion matrices for baselines without the MKD procedure. The bottom row is the confusion matrices when adding MKD.

A.2. Last layer Bias

In Table 7 we share extra experiments regarding the impact of the NCM trick. Specifically, OCM and DVC are not included in the main paper.

Method	Logits Acc.	NCM Acc.
ER [NeurIPS‘19]	$23.0_{\pm 0.8}$	$29.0_{\pm 0.3}$ (↑6.0)
ER + ours	$38.5_{\pm 0.5}$	$31.5_{\pm 0.5}$ (↓7.0)
DER++ [NeurIPS‘20]	$22.8_{\pm 1.8}$	$26.6_{\pm 3.4}$ (↑3.8)
DER++ + ours	$32.1_{\pm 0.5}$	$28.7_{\pm 1.4}$ (↓3.4)
DVC [CVPR‘22]	$19.5_{\pm 2.1}$	$21.1_{\pm 1.5}$ (↑1.6)
DVC + ours	$27.0_{\pm 2.0}$	$27.4_{\pm 1.7}$ (↑0.4)
ERACE [ICLR‘22]	$27.4_{\pm 0.6}$	$28.1_{\pm 0.7}$ (↑0.7)
ERACE + ours	$37.0_{\pm 0.7}$	$34.2_{\pm 0.2}$ (↓2.8)
OCM [ICML‘22]	$29.1_{\pm 1.4}$	$29.3_{\pm 1.2}$ (↑0.2)
OCM + ours	$37.1_{\pm 0.7}$	$31.6_{\pm 0.2}$ (↓5.5)
GSA [CVPR‘23]	$29.7_{\pm 1.5}$	$32.6_{\pm 1.6}$ (↑2.9)
GSA + ours	$40.1_{\pm 1.0}$	$36.2_{\pm 0.5}$ (↓3.9)

Table 7. Final Average Accuracy (%) on CIFAR100 M=1k of various baselines, with and without using the NCM trick. Logits Acc. refers to the accuracy of the model using predicted logits while NCM Acc. refers to NCM accuracy trained on intermediate representations from memory at the end of training.

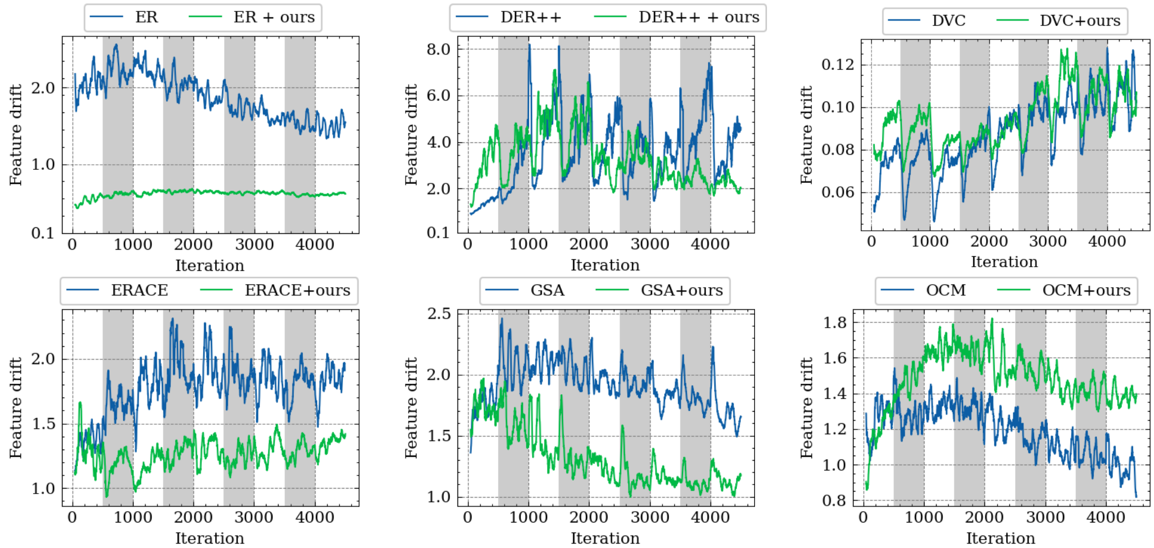


Figure 10. Feature drift d_t of *ER*, *DER++*, *DVC*, *ERACE*, *GSA*, *OCM* and their MKD adaptations on CIFAR100, M=5k.

A.3. Feature Drift

We show additional experiments concerning the impact of MKD on feature drift on Figure 10. It can be observed that for *GSA* and *ERACE*, introducing MKD can greatly help in reducing feature drift. However, this phenomenon is not as pronounced with *DVC* and *DER++*. Since *DVC* encourages representations to be augmentation-invariant, it is expected to observe more stability against feature drift with *DVC*. Notably, the drift values of *DVC* and *DVC + ours* are considerably lower than any other considered method. Additionaly, we observe the opposite effect for OCM, which also incorporate feature stability by leveraging a contrastive objective (Guo et al., 2022). Even though MKD cannot reduce feature drift for OCM, experimental results still demonstrate a significant improvement in performances.

A.4. Feature Discrimination

To showcase the impact of MKD on feature discrimination, we presented t-SNE results on memory data at the end of training for *ER* and *ER + ours*. In Figure 11 we present additional t-SNE experiments for remaining baselines. We used a perplexity of 30 for these experiments.

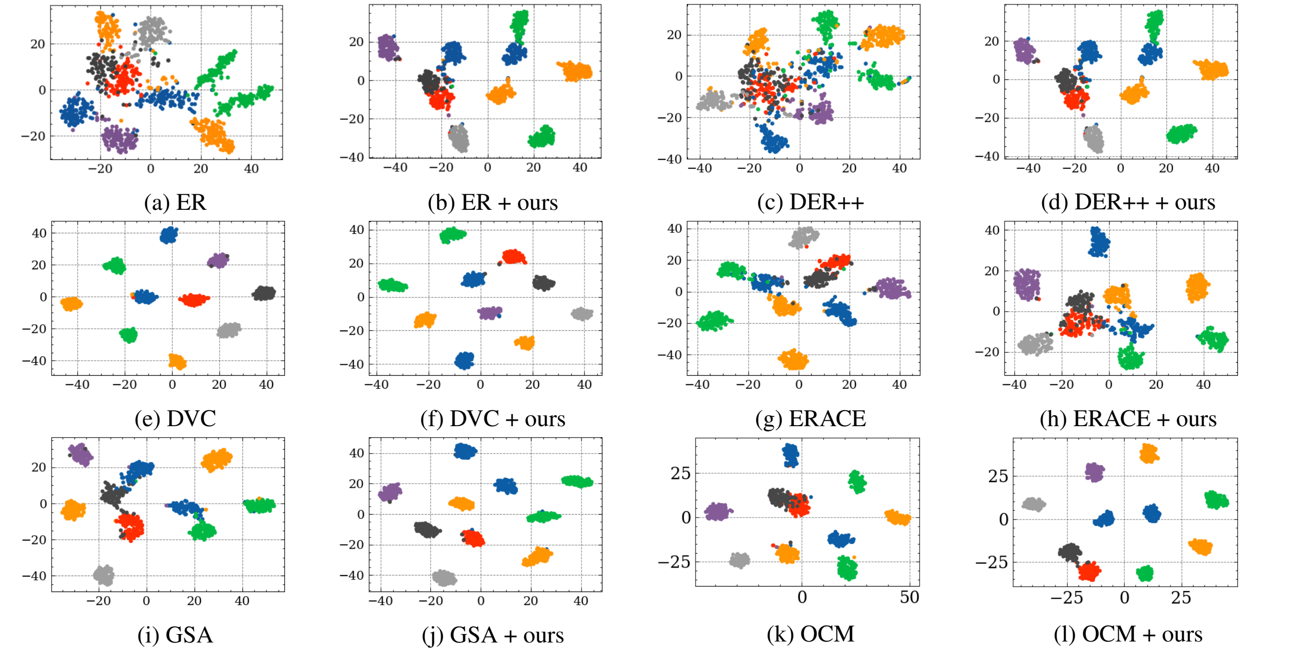
A.5. Backward Transfer

In Table 8 we present additional experiments concerning the impact of MKD on Backward Transfer (BT). Specifically, OCM and DVC are not included in the main paper because of the limited space.

B. Experimental Details

B.1. Datasets

We use variations of standard image classification datasets (Krizhevsky, 2009; Le & Yang, 2015; Deng et al., 2009). The original datasets are split into several tasks of non-overlapping classes. Specifically, we experimented on CIFAR10, CIFAR100, Tiny ImageNet, and ImageNet-100.
CIFAR10 contains 50,000 32x32 train images and 10,000 test images and is split into 5 tasks, each containing 2 classes, for a total of 10 distinct classes.
CIFAR100 contains 50,000 32x32 train images and 10,000 test images and is split into 10 tasks, each contains 10 classes, for a total of 100 distinct classes.
Tiny ImageNet is a subset of the ILSVRC-2012 classification dataset and contains 100,000 64x64 train images as well as 10,000 test images and is split into 20 tasks, each containing 10 classes, for a total of 200 distinct classes.
ImageNet-100 is another subset of ILSVRC-2012 containing only the first 100 classes with 1,300 224x224 images per class



(a) ER (b) ER + ours (c) DER++ (d) DER++ + ours

(e) DVC (f) DVC + ours (g) ERACE (h) ERACE + ours

(i) GSA (j) GSA + ours (k) OCM (l) OCM + ours

Figure 11. t-SNE visualization of *ER*, *DER++*, *DVC*, *ERACE*, *GSA*, *OCM* and their MKD adaptations on CIFAR100, M=5k.

Method	CIFAR100	ImageNet100
ER	-16.7±1.2	-17.5±1.5
ER + ours	**+8.15±0.8**	**-1.3±2.3**
DER++	-27.5±3.4	-18.9±2.5
DER++ + ours	**-10.4±5.6**	**-14.4±2.5**
ERACE	-3.0±1.6	-6.5±1.3
ERACE + ours	**+8.6±1.6**	**-5.6±1.4**
DVC	-32.5±4.7	-34.3±5.9
DVC + ours	**-20.9±6.6**	**-31.5±0.8**
OCM	-5.0±3.0	+1.1±1.6
OCM + ours	**+14±1.8**	**+3.9±0.8**
GSA	-4.9±1.2	-17.0±1.3
GSA + ours	**-2.5±3.1**	**-15.5±1.0**

Table 8. Backward Transfer (%) at the end of training on CIFAR100, M=5k and Imagenet100, M=10k for various baselines. Higher is better. Mean and standard deviations over 5 runs are displayed.

for training and 50 for testing.

B.2. Data Augmentation

Several methods have demonstrated improved performance through the use of simple augmentations rather than more intricate ones. To ensure optimal performance comparison among the various methods, we employed two distinct augmentation strategies: the *partial* and the *full* strategies.

***Partial* Augmentation Strategy.** The *partial* augmentation strategy comprises only a subset of the augmentations utilized in the *full* strategy. Specifically, it involves a sequence of random cropping and random horizontal flipping, both with a probability p of 0.5.

***Full* Augmentation Strategy.** The *full* augmentation strategy encompasses a wider array of augmentations. It involves a sequence of random cropping, horizontal flipping, color jitter, and random grayscale transformations. The parameters for color jitter are set to $(0.4, 0.4, 0.4, 0.1)$ with a probability p of 0.8. The application probability for random grayscale is set at 0.2.

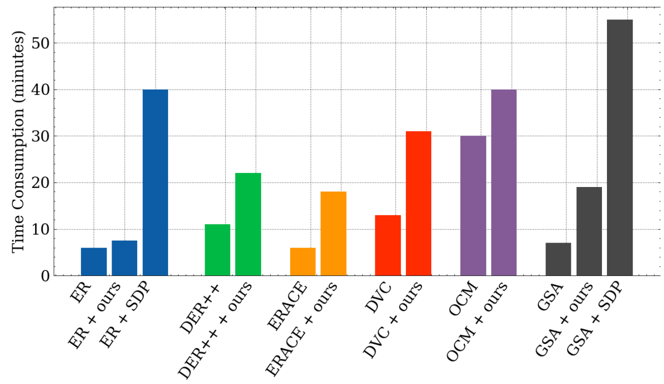


Figure 12. Time consumption (minutes) of compared methods when training on CIFAR100, M=5k with V100 GPUs.

These strategies have also been chosen during the hyper-parameter search.

B.3. Task boundaries inference

For experimenting on the *blurry* setting with OCM (Guo et al., 2022), it is necessary to infer the task change. Inferring task change in this setup can be cumbersome and grandly impact performances. For simplicity, we detect task change by applying two simple rules. We consider the task has changed if:

- A new class (never seen by the model) appears in the stream;
- The last task change appeared at least 100 iterations previous to the current one.

B.4. Hyper-parameters table

Different hyper-parameters values used in grid search for considered methods are reported in Table 9. This grid search has been conducted on CIFAR100, M=5k. Note that we used parameters from the original paper for OCM (Guo et al., 2022) due to computational constraints.

B.5. Hardware and computation

For the compared methods, we trained on RTX A5000 and V100 GPUs. Figure 12 references the training time of each method on CIFAR100 M=5k.

Method	Parameter	Values
ER	optim	[SGD, Adam]
	weight decay	[0, 1e-4]
	lr	[0.0001, 0.001, 0.01, 0.1]
	momentum	[0, 0.9]
	aug. strat.	[full, partial]
ER-ACE	optim	[SGD, Adam]
	weight decay	[0, 1e-4]
	lr	[0.0001, 0.001, 0.01, 0.1]
	momentum	[0, 0.9]
	aug. strat.	[full, partial]
DER++	optim	[SGD, Adam]
	weight decay	[0, 1e-4]
	lr	[0.0001, 0.001, 0.01, 0.03]
	momentum	[0, 0.9]
	aug. strat.	[full, partial]
	alpha	[0.1, 0.2, 0.5, 1.0]
	beta	[0.5, 1.0]
DVC	optim	[SGD, Adam]
	weight decay	[0, 1e-4]
	lr	[0.0001, 0.001, 0.01, 0.1]
	momentum	[0, 0.9]
	aug. strat.	[full, partial]
GSA	optim	[SGD, Adam]
	weight decay	[0]
	lr	[0.0001, 0.0005, 0.01, 0.05, 0.01]
	momentum	[0]
	aug. strat.	[full, partial]
ER+SDP	optim	[Adam]
	weight decay	[0]
	lr	[0.0003]
	momentum	[0]
	μ	[10, 100, 1000, 10000]
	c^2	[0.5, 0.75, 0.9]
PCR	optim	[Adam]
	weight decay	[0]
	lr	[0.0005]
	momentum	[0.9]
	aug. strat.	[full]

Table 9. Hyper-parameters tested for every method on CIFAR100, M=5k, 10 tasks.