Orchestrate Latent Expertise: Advancing Online Continual Learning with Multi-Level Supervision and Reverse Self-Distillation

Hongwei Yan[1] Liyuan Wang[2*] Kaisheng Ma[3*] Yi Zhong[1*]

[1] School of Life Sciences, IDG/McGovern Institute for Brain Research, Tsinghua University
[2] Dept. of Comp. Sci. & Tech., Institute for AI, BNRist Center,
Tsinghua-Bosch Joint ML Center, THBI Lab, Tsinghua University
[3] Institute for Interdisciplinary Information Sciences, Tsinghua University

yanhw22@mails.tsinghua.edu.cn, {wly2023, kaisheng, zhongyithu}@tsinghua.edu.cn

Abstract

To accommodate real-world dynamics, artificial intelligence systems need to cope with sequentially arriving content in an online manner. Beyond regular Continual Learning (CL) attempting to address catastrophic forgetting with offline training of each task, Online Continual Learning (OCL) is a more challenging yet realistic setting that performs CL in a one-pass data stream. Current OCL methods primarily rely on memory replay of old training samples. However, a notable gap from CL to OCL stems from the additional overfitting-underfitting dilemma associated with the use of rehearsal buffers: the inadequate learning of new training samples (underfitting) and the repeated learning of a few old training samples (overfitting). To this end, we introduce a novel approach, Multi-level Online Sequential Experts (MOSE), which cultivates the model as stacked sub-experts, integrating multi-level supervision and reverse self-distillation. Supervision signals across multiple stages facilitate appropriate convergence of the new task while gathering various strengths from experts by knowledge distillation mitigates the performance decline of old tasks. MOSE demonstrates remarkable efficacy in learning new samples and preserving past knowledge through multi-level experts, thereby significantly advancing OCL performance over state-of-the-art baselines (e.g., up to 7.3% on Split CIFAR-100 and 6.1% on Split Tiny-ImageNet). [†]

1. Introduction

Continual Learning (CL) aims to improve the adaptability of AI systems to the ever-evolving environment of the real world [35, 42, 47, 66, 74]. In this pursuit, Online Continual Learning (OCL) has emerged as an important paradigm that

---

[*]Corresponding authors.
[†]Our code is available at https://github.com/AnAppleCore/MOSE

mirrors realistic scenarios with one-pass data streams and attracted a wide range of interest in related fields [24, 41, 52]. However, OCL also faces serious challenges such as efficient online operation, limited input data, and stringent resource constraints, which remain to be solved.

In this work, we first analyze the unique requirements of OCL compared to regular CL: the training samples for each task are encountered only once, which makes the model susceptible to inadequate learning of each task [31]. Saving samples of the current task in the buffer could partially solve it, but overstepping the training of buffered data will trigger a severe forgetting problem, namely, overfitting to the memory buffer for old tasks [57, 73]. This critical issue is attributed to the **overfitting-underfitting dilemma** across data distributions of new and old tasks, which specifies the distinction between CL and OCL. While recent strides have been made with the help of data augmentation [73, 75] and contrastive representation learning [12, 38, 40, 71], shortcomings such as the sub-optimality of convergence and the demand for large batch size with higher computational overhead remind us a chasm in performance persists [52].

Unlike AI systems, the biological brain has evolved an innate capacity for continual learning in an online manner. In particular, the mammalian visual processing system extracts multi-level features of online inputs and memorizes them for further reuse [19, 29, 33, 55]. Neurons in these areas not only process and transmit signals progressively but also communicate across cortex levels through diverse neural circuits [14]. The *Shallow Brain* hypothesis [53] is proposed to describe the ability of the superficial cortex to work independently and cooperate with deeper ones. Inspired by this, we propose the **M**ulti-level **O**nline **S**equential **E**xperts (MOSE), a novel approach to drive the leap forward in OCL. MOSE consists of two major components: multi-level supervision and reverse self-distillation. The former empowers the model to forge hierarchical features across

various scales, cultivating the continual learner as stacked sub-experts excelling at different tasks. Meanwhile, the latter shifts knowledge within the model from shallower experts to the final predictor, gathering the essence of diverse expertise. Combined with both modifications, the model tackles tasks by orchestrating a harmonious symphony of latent network skills, endowed with resistance against distribution shift and task-wise accuracy.

To address the identified particular challenge of OCL, MOSE places the model at an appropriate convergence point to facilitate efficient learning of new tasks, while avoiding performance degradation of old tasks. The cooperation of multi-level experts achieves such a flexible balance between overfitting and underfitting. As a result, our MOSE substantially outperforms state-of-the-art baselines.

Our contributions can be summarized as three aspects:
1. We present an extensive analysis of the OCL problem and attribute its particular challenge to the overfitting-underfitting dilemma of the observed data distributions.
2. We propose an innovative approach with multi-level supervision and reverse self-distillation, to achieve appropriate convergence in an online manner.
3. Empirical experiments demonstrate the superior performance of MOSE over state-of-the-art baselines.

2. Related Work

Continual Learning (CL). CL has received increasing attention in recent years [42, 66, 74], characterized by non-stationary data learning. Conventionally, CL methods are classified into three groups: architecture-, regularization- and replay-based [35, 47, 51]. Architecture-based methods focus on allocating dedicated parameter subspace, including parameter isolation [50, 68], dynamic architecture [17, 46, 59, 65], and modular network [62, 64]. Regularization-based methods [26, 32, 39, 61] mitigate catastrophic forgetting by introducing explicit regularization terms to balance new and old tasks. Replay-based methods [6, 9, 48, 60, 63] exploit an additional memory buffer to save a subset of old training samples, to recover previous data distributions. As regular CL methods usually perform multi-epoch training of each task, it remains extremely challenging to deal with the one-pass data stream in OCL [21].

Online Continual Learning (OCL). In OCL, a model needs to learn a one-pass data stream with shifting distribution [24, 41, 52]. Replay-based methods have been extensively explored in OCL thanks to their efficacy and generality [52]. ER [10] applies a reservoir sampling strategy [58] and randomly updates the memory buffer. MIR [3] selects the most interfered replay samples. SCR [40] takes supervised contrastive loss and nearest-class-mean classifier. ER-AML [7] modifies cross-entropy loss to mitigate representation drift. OCM [21] learns holistic features through maximizing mutual information. OnPro [67] avoids short-

cut learning using online prototypes. GSA [22] improves cross-task discrimination using a gradient-based method. Specifically, we adopt data augmentation for its capacity to expand the data distribution across both stored and incoming batches. The combination of *memory replay* and *data augmentation* currently dominates the OCL literature, yet architecture-based methods remain to be explored in OCL. Our method integrates their strengths and is compatible with mainstream replay-based methods.

Knowledge Distillation (KD) in CL. KD [5, 27] usually saves a frozen copy of the old model to "teach" the current model. According to the target space, there usually exist logits-based, feature-based, and relation-based KD methods [74]. LwF [37] learns outputs from the previous model; DER [6] employs logit-based distillation, where previous logits are saved in memory buffer; iCaRL [49] preserves learned representations through distillation; CO^2L [8] employs self-supervised distillation to keep robust representations; MuFAN [31] introduces structure-wise distillation to build relation between tasks and CCPR [71] transfers feature correlation between training samples from past model. In contrast, our method innovates by implementing KD internally within the current model architecture, setting it apart from conventional KD techniques.

3. Preliminaries

3.1. Notation and Setups

Problem Description. Let's consider a general setting of OCL, which usually refers to fitting a one-pass data stream with non-stationary distribution [24, 41, 52]. The training dataset $\mathcal{D}$ is split into several tasks, $\mathcal{D}_{t \leq T} = \{(x_i, y_i)|y_i \in \mathcal{C}_t\}$, and $\bigcup_{t=1}^{T} \mathcal{C}_t = \mathcal{C}$ is the set of labels, where the sets of class labels are disjoint $\mathcal{C}_t \cap \mathcal{C}_{t'} = \emptyset$. The continual part requires that task training sets arrive sequentially and are not accessible without additional cost, whenever the training phase of the corresponding task is complete. The unavailability of past data causes a biased input distribution and therefore induces catastrophic forgetting [16, 18, 43] of old tasks. This challenge is further strengthened for OCL: during the training phase of task t, sequential data batches $\mathcal{B}^t = \{(x_i, y_i)\}_{i=1}^{B}$ of batch size B are sampled from its training set $\mathcal{D}_t$ in a no replacement manner. Models are restricted to process each training sample (input-label pair) precisely once – **at most one epoch over each training set**. Thus, the continual learner also struggles to learn each new task. To tackle this challenge, OCL methods frequently employ memory replay strategy [21, 67]. They apply a memory buffer $\mathcal{M}$ with limited capacity M to store old training samples. At each training iteration, a batch of data $\mathcal{B}^{\mathcal{M}} = \{(x_i, y_i)\}_{i=1}^{B^{\mathcal{M}}}$ of batch size $B^{\mathcal{M}}$ is retrieved from the memory buffer $\mathcal{M}$, and the model is updated according to the loss calculated over $\mathcal{B}^t$ and $\mathcal{B}^{\mathcal{M}}$ (detailed in Sec. 3.2).

Model Architecture. Implemented as multiple consecutive modules, deep learning models depend heavily on the power of network depth. Backbones commonly used in computer vision like ResNet [25] and ViT [56] consist of stacked repetitive blocks such as convolution and multi-headed self-attention. A general network $F : \mathbb{R}^m \mapsto \mathbb{R}^{|\mathcal{C}|}$ maps an input x of size m to a class probability vector $\hat{y}$ for classification problem, consisting of a feature extractor f_θ and an output layer g_ϕ with parameters θ, ϕ. The feature extractor f_θ includes a series of successively connected blocks and encodes x to a feature vector of dimension d:

$$f_\theta(x) = \left(f_{\theta_n} \circ f_{\theta_{n-1}} \circ \cdots \circ f_{\theta_1}\right)(x) \in \mathbb{R}^d,$$
$$\theta = \{\theta_1, \theta_2, \cdots, \theta_n\}, \quad (1)$$

where each block f_{θ_i} takes the feature map h_{i-1} from its predecessor $f_{\theta_{i-1}}$ and produces a new one $h_i = f_{\theta_i}(h_{i-1})$. Multi-level feature maps are obtained from blocks with different depths ($h_0 = x$, $h_n = f_\theta(x)$). And a classifier network learns to minimize the classification loss $\mathcal{L}(\hat{y}, y)$ between its output $\hat{y} = F(x; \theta, \phi) = g_\phi(h_n) = g_\phi\left(f_\theta(x)\right)$ and ground-truth y. Cross-entropy loss is usually used:

$$\mathcal{L}_{ce}(\hat{y}, y) = -\sum_{c \in \mathcal{C}} y^c \log\left(\frac{\exp(\hat{y}^c)}{\sum_{s \in \mathcal{C}} \exp(\hat{y}^s)}\right). \quad (2)$$

y^c and $\hat{y}^c$ are the probabilities of being classified as c.

3.2. Representative Method

Regarding the memory buffer $\mathcal{M}$, three aspects need to be considered: (1) **memory buffer size** M to determine the number of old training samples that can be stored in the memory buffer; (2) **memory retrieval strategy** [3, 51] to gather old training samples from the memory buffer; and (3) **memory update strategy** [4, 23, 30] to keep an appropriate data structure across tasks and classes. Existing methods [4, 13, 30, 71] demonstrate variability reflecting diverse interpretations of the OCL problem and numerous possible solutions. Here we describe two representative baselines:
Experience Replay (ER). The most straightforward baseline is the naive ER [10], using reservoir sampling strategy [58] to randomly retrieve old samples from the buffer and also update it with random addition and deletion in a balanced manner. The model F learns incremental batches and old training samples $\mathcal{B} = \mathcal{B}^{\mathcal{M}} \cup \mathcal{B}^t$ with cross-entropy:

$$\mathcal{L}_{\text{ER}} = \mathbb{E}_{(x_i, y_i) \in \mathcal{B}} \mathcal{L}_{ce}\left(F(x_i; \theta, \phi), y_i\right). \quad (3)$$

Supervised Contrastive Replay (SCR). SCR [40] introduces a supervised contrastive loss that learns more refined feature encoding by drawing the same class closer while pushing different classes away. A nearest-class-mean (NCM [44]) classifier is used instead of output class probability directly. During the testing phase, SCR compares the distance between the encoded input feature with class feature means and predicts the closest one. Replacing g_ϕ with

a linear projection layer p_ψ, SCR computes its loss $\mathcal{L}_{scl}$ between features $q = F(x; \theta, \psi) = p_\psi\left(f_\theta(x)\right)$ over $\mathcal{B}$:

$$\mathcal{L}_{scl}(q_i, y_i) = -\mathbb{E}_{p \in \mathcal{P}(i)} \log\left(\frac{\exp(q_i \cdot q_p / \tau)}{\sum_{j \neq i} \exp(q_i \cdot q_j / \tau)}\right), \quad (4)$$
$$\mathcal{L}_{\text{SCR}} = \mathbb{E}_{(x_i, y_i) \in \mathcal{B}} \mathcal{L}_{scl}(F(x_i; \theta, \psi), y_i),$$

where τ is a temperature hyperparameter, and $\mathcal{P}(i)$ is the index set of positive samples in $\mathcal{B}$, whose class labels are the same as x_i but excluding x_i itself.

3.3. Empirical Analysis

The significant change in data distribution and inadequate training for each current task jointly constitute the central challenge of OCL. Although memory replay is helpful, its assistance is quite limited and costly. For the later issue, previous works [73] point out that, updating multiple times over one incoming batch directly is not an ideal choice because it gradually becomes biased on this small batch. Data augmentation could enhance the unified data distribution of memory buffer and incoming batches. However, repetitive training still unavoidably brings in much more calculation, and therefore to a certain extent it violates the online requirements of efficiency [73]. Although contrastive-based representation learning methods exhibit more generalization ability [20, 38, 40, 71], the scarcity of available training samples in OCL keeps them from optimal convergence.

To further study the impact of training iterations and data augmentation on the OCL problem, aiming to narrow the performance gap between OCL and offline CL, we conducted a toy experiment to assess the task performance of ER and SCR under different numbers of task epochs and augmentation conditions (see Fig. 1). We evaluate both task-wise prediction performance and the average accuracy right after completing the second task of Split CIFAR-100 [34], denoted by $t = 2$. The results at $\text{epoch} = 1$ correspond to OCL, while offline CL has no specified upper limit on epochs. As epochs increase, the training process in OCL is closer to that in offline CL.

Fig. 1 represents the insufficient training of ER over new task: with more epochs of learning, ER performs better with the current task (Fig. 1b). However, the descending curve in Fig. 1a and flat curve in Fig. 1c of ER indicate the improvement of task 2 sacrifices the accuracy of task 1. This balance issue is partially resolved by data augmentation (ER+aug): both new and old tasks gain promotion as the epoch number grows. On the other hand, we capture a significant disparity between fewer and more epochs in Fig. 1c (augmentation operation raises the upper limit), showing its capability to ascend to a superior level. Contrastive learning slightly reduced this gap (similar accuracy as ER+aug at epoch=8, but higher starting point at epoch=1) and exhibits better stability (SCR+aug in Fig. 1a, 1c), yet still far from optimal.

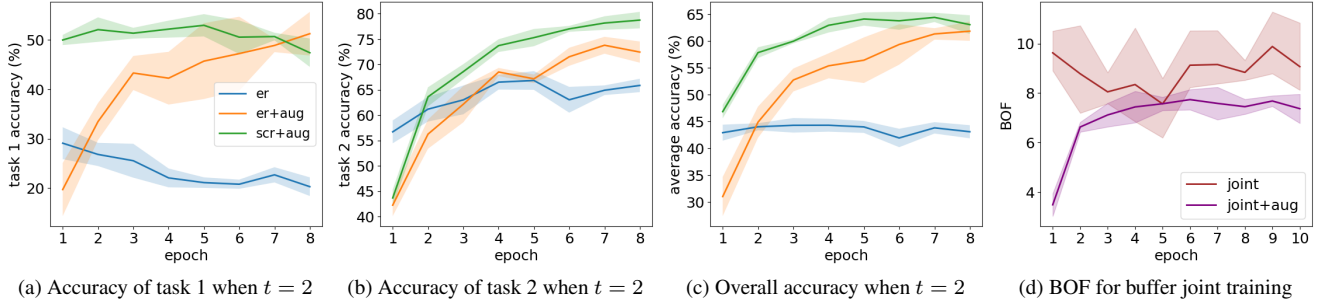| (a) Accuracy of task 1 when $t = 2$ | (b) Accuracy of task 2 when $t = 2$ | (c) Overall accuracy when $t = 2$ | (d) BOF for buffer joint training |

Figure 1. **Overfitting-Underfitting Dilemma.** We show the impact of training different epochs to the test accuracy of task 1 and task 2 of Split CIFAR-100 [34] dataset, as well as BOF value for joint training on the buffer of the last task. **(a)** shows the test accuracy of task 1 when the training of task 2 has just finished ($t = 2$). Similarly, **(b)** and **(c)** show the test accuracy of task 2, and the average performance of the first two tasks when $t = 2$. **(d)** shows the buffer overfitting problem across different epochs. aug here is a combination of 3 different data augmentation used in [11, 21]. For a fair comparison, we fix $M = 1K$, batch size $B = 10$, and buffer batch size $B^{\mathcal{M}} = 64$.

Furthermore, it is important to note that, in replay-based methods, buffered data serves as the core mechanism for preserving past knowledge. Those data can be replayed multiple times throughout the training process. (*e.g.*, for an OCL benchmark with 10 tasks, buffered data from task 1 will be retrieved for 9 epochs during subsequent training), so we conduct another toy example to evaluate the generalization gap between the buffer and the test set. Specifically, we train ER only on the buffer of task 10 (contains a balanced number of samples for each class) for different epochs (denoted as joint, see Fig. 1d). We design a new metric Buffer Overfitting Factor (**BOF**) to quantify:

$$\text{BOF} = \frac{\text{Buffer accuracy} - \text{Test set accuracy}}{\text{Test set accuracy}}. \quad (5)$$

The difference in accuracy between the buffer and the test set is normalized by the test set accuracy, providing a measure of overfitting in relative terms for equitable assessment. The lines in Fig.1d reveal that even with augmentation, replaying exemplars from a fixed buffer only twice (epoch=2) yields a substantial BOF. As epochs increase, the BOF stabilizes, indicating a pronounced overfitting phenomenon, similar to the non-augmented case.

Overfitting-Underfitting Dilemma. Based on the above analysis, we pinpoint the inherent challenge of OCL: OCL methods must navigate the underfitting of the current task while simultaneously avoiding overfitting to the buffered data of previous tasks. This dual challenge inspires us to introduce a novel approach, as outlined below.

4. Method

Inspired by the hierarchical advantages of the mammalian visual processing system [15, 19, 29, 33, 55, 70], we propose to inject multi-level supervision in our continual model and aggregate the skills from different experts within it (see Fig. 2), which includes two synergistic components described in Sec. 4.1 and Sec. 4.2, respectively. Then we summarize the overall framework in Sec. 4.3.

4.1. Multi-Level Supervision

As formulated in Sec. 3.1, we consider deep neural networks as a composition of sequentially connected building blocks: $f_{\theta_n} \circ f_{\theta_{n-1}} \circ \cdots \circ f_{\theta_1}$. The input data is encoded and propagated through the network layer by layer, producing hierarchical feature maps for each training sample. Previous research [31] suggested that multi-scale feature maps benefit vision tasks by providing multi-level information ranging from pattern-wise knowledge to highly semantic understanding. This further motivates us to aggregate latent expertise from shallower layers to deeper ones.

First, the whole network can be split into n blocks according to the backbone architecture. Corresponding feature maps $\{h_i \in \mathbb{R}^{d_i}\}_{i \leq n}$ are obtained through forward data flow $h_i = f_{\theta_i}(h_{i-1})$, yet they lie in embedding spaces with different dimension $\{d_i\}_{i \leq n}$, ($d_0 = m$ is the input size, and $d_n = d$ is the size of final feature to be fed into output head g_ϕ or p_ψ). To unify the feature size for further calculation and comparison, we introduce a dimension alignment module $a_{\omega_i} : \mathbb{R}^{d_i} \mapsto \mathbb{R}^d$ with parameter ω_i to project each feature map to the space with same dimension as the last feature map [72], namely, $\hat{h}_i = a_{\omega_i}(h_i) \in \mathbb{R}^d$. These alignment modules serve a dual purpose: they support a general platform for vector operations within hidden space and extract meaningful, diverse information from feature maps that are comparatively sparse and less plausible.

Latent Sequential Experts. Based on the above dimension alignment of multi-scale features, it becomes possible to train each block of the network as a fully functional continual learner. To do so, we add output heads after each block to transform the projected feature $\hat{h}_i$ into an output vector for supervised loss computation. The choice of output head depends on the supervision loss we use, and this multi-level framework is generally compatible with most replay-based methods. Shown in Fig. 2, we add two types of output heads after all alignment module a_{ω_i}: p_{ψ_i} and g_{ϕ_i} for supervised contrastive representation learning and cross-entropy-based
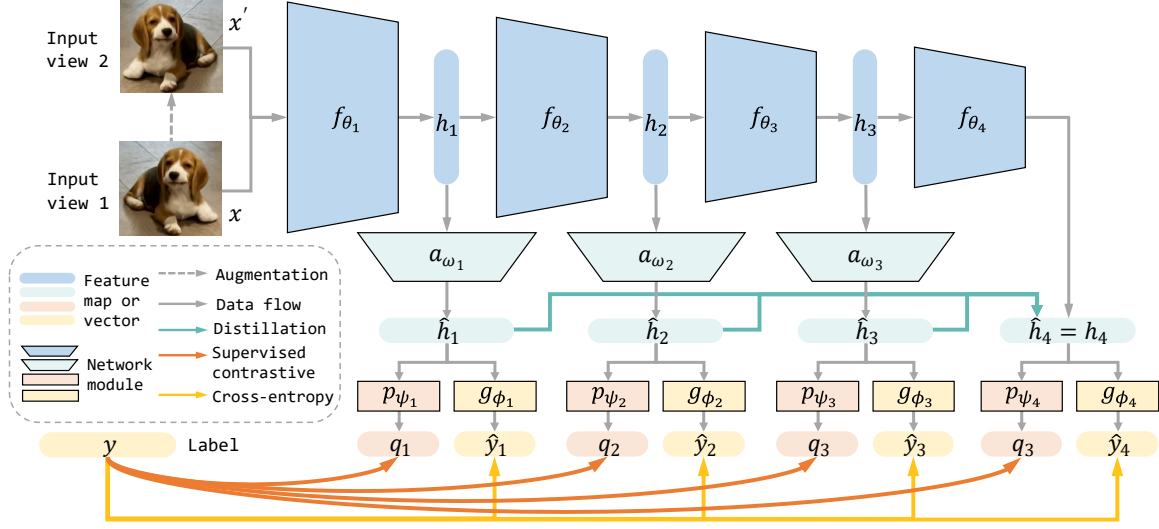
Figure 2. **Illustration of the proposed MOSE.** For each training sample (x, y), the input x is augmented to another view x' and concatenate together for network training. MOSE includes multiple supervision signals (cross-entropy and supervised contrastive loss) injected at different network layers and extra reverse self-distillation from the shallower layers to the deepest to integrate the knowledge of experts.

classification learning (see Sec. 3.2), respectively. We then mark them as latent experts E_i, consist of sequential blocks $\{f_{\theta_j}\}_{j \leq i}$ up to block i and corresponding feature alignment module a_{ω_i} and output heads p_{ψ_i} and g_{ϕ_i}. Naturally, each expert is trained with compounded supervision loss $\mathcal{L}_{E_i}$:

$$\mathcal{L}_{E_i}(x, y) = \mathcal{L}_{ce}(\hat{y}_i, y) + \mathcal{L}_{scl}(q_i, y),$$
$$\text{where, } \hat{y}_i = E_i(x; \theta_{1:i}, \phi) = g_{\phi_i}(a_{\omega_i}(f_{\theta_{1:i}}(x))), \quad (6)$$
$$q_i = E_i(x; \theta_{1:i}, \psi) = p_{\psi_i}(a_{\omega_i}(f_{\theta_{1:i}}(x))).$$

Therefore, **M**ulti-**L**evel **S**upervision (**MLS**) signal $\mathcal{L}_{\text{MLS}}$ is injected into each block of the network by summing up expert-wise losses and this framework fits perfectly with most replay-based methods:

$$\mathcal{L}_{\text{MLS}} = \mathbb{E}_{(x_i, y_i) \in \mathcal{B}} \sum_{j=i}^{n} \mathcal{L}_{E_j}(x_i, y_i). \quad (7)$$

Separate $\mathcal{L}_{ce}$ for New Task. Recent research indicates that the gradient unbalance observed in output logits of models trained with cross-entropy will lead to bad decision boundaries between old and new tasks [1, 7, 22]. From this perspective, we revise the cross-entropy loss $\mathcal{L}_{ce}$ used in expert's loss Eq. 6 as $\mathcal{L}_{ce} = \mathcal{L}_{ce,\text{new}} + \mathcal{L}_{ce,\text{buf}}$ to avoid severe gradient bias caused by repeatedly calculating $\mathcal{L}_{ce}$ [1]. specifically, $\mathcal{L}_{ce,\text{buf}}$ is normally calculated over buffer batch $\mathcal{B}^{\mathcal{M}}$, based on the assumption that memory buffer $\mathcal{M}$ exhibits a better balance across tasks and classes. For incoming batch $\mathcal{B}^t$, loss computed over output logits that represent classes contained in current task $\mathcal{C}^t$:

$$\mathcal{L}_{ce,\text{new}}(\hat{y}, y) = \sum_{c \in \mathcal{C}^t} y^c \log\left(\frac{\exp(\hat{y}^c)}{\sum_{s \in \mathcal{C}^t} \exp(\hat{y}^s)}\right). \quad (8)$$

4.2. Reverse Self-Distillation

Mixture-of-expert is an effective strategy in machine learning, which cooperates diversified skills of different experts to achieve better predictions [2, 54, 62]. A consistent phenomenon is observed when we test the task accuracy for each expert E_i. In Tab. 1, the task-wise accuracy of experts, when equipped with MLS, is recorded. Under an ideal situation, we assume that the network can choose the best expert for each task, and the maximum value denoted as *MAX* stands for the upper limit of output from a single expert. We also calculate the accuracy of the average output logits from all experts as a trivial integration process termed as *MOE*.

Based on the result of the first 6 rows of Tab. 1, we conclude experts show differences across tasks, and utilizing multiple experts' outputs surpasses the accuracy of any single expert even if we select the most appropriate one for each task. This further proves the importance of knowledge aggregation across sub-learners. But MOE comes with additional computation and storage costs, and it is not optimal when we have introduced auxiliary modules like feature alignment modules a_ω and extra output heads p_ψ, g_ϕ. Therefore, we intend to transfer skills from different experts to the last one E_4, which is indeed the whole network F itself, to avoid redundant overhead at the testing phase.

For this purpose, we implement a novel distillation process called **R**everse **S**elf-**D**istillation (**RSD**). Different from the traditional self-distillation [72], our RSD takes latent sequential experts as teachers and treats the largest expert F as the student. It calculates the L2 distance between normalized feature $\hat{h}_i$ (detached to stop gradient back-propagation) of each expert $E_{i<n}$ and the final feature h_n of E_n (Eq. 9).

Experts		task 0	task 1	task 2	task 3	task 4	task 5	task 6	task 7	task 8	task 9	Average
	E_1	**50.54**	42.85	51.54	41.55	46.36	50.33	48.43	45.51	50.70	50.16	47.80
	E_2	47.06	**45.95**	54.47	45.27	50.28	53.87	51.89	50.59	54.70	54.41	50.85
	E_3	50.17	45.87	**55.42**	**45.55**	**50.89**	**55.23**	**53.54**	**53.97**	**59.69**	61.73	**53.21**
w/o RSD	$E_4 = F$	49.70	44.43	54.08	43.95	49.71	53.72	52.78	53.41	59.41	**65.15**	52.63
	MAX	50.54	45.95	55.42	45.55	50.89	55.23	53.54	53.97	59.69	65.15	53.59
	MOE	52.99	48.47	58.28	47.63	53.59	57.33	55.76	56.13	62.78	64.23	55.72
	E_1	46.79	43.33	50.74	41.19	46.11	49.80	48.20	45.27	50.02	49.66	47.11
	E_2	50.83	46.67	54.11	44.55	49.45	53.85	52.29	49.50	54.75	54.21	51.02
	E_3	51.67	47.10	55.39	45.93	51.08	54.79	54.51	53.36	59.71	61.13	53.47
w/ RSD	$E_4 = F$	**52.31**	**48.87**	**56.75**	**47.28**	**51.79**	**55.72**	**54.93**	**54.38**	**61.45**	**62.26**	**54.57**
	MAX	52.31	48.87	56.75	47.28	51.79	55.72	54.93	54.38	61.45	62.26	54.57
	MOE	53.96	50.29	57.95	48.29	53.31	57.11	55.75	55.23	61.90	62.15	55.59

Table 1. **Task-wise End Accuracy (higher is better)** assessed over Split CIFAR-100 [34] dataset with memory size $M = 5000$. We record the mean result (%) from 15 runs, and highlight the highest accuracy for each task among 4 sub-experts with **bold** font. Here $E_4 = F$ represents the entire network. *MAX* calculates the ideal situation where we choose the expert with maximal precision for each task, and *MOE* stands for the accuracy of averaged output logits across all experts. We choose NCM classifier [44] for all experts.

The introduction of RSD brings a positive effect on OCL, as shown in Tab. 1. First, it successfully helps the last expert become the strongest one among all experts without weakening any of them as well as the MOE version. Second, the last expert makes progress on tasks where its initial performance is lacking, contributing positively to OCL challenges. Additional insights on RSD with alternate experts serving as students are detailed in Appendix 9.1.

$$\mathcal{L}_{\text{RSD}} = \mathbb{E}_{(x_i, y_i) \in \mathcal{B}} \sum_{i=1}^{n-1} \left\| \hat{h}_i' - h_n' \right\|_2, \tag{9}$$

$$h' = \text{normalize}(h) = h/\|h\|_2.$$

4.3. Overall Framework of MOSE

Put MLS and RSD together, we have established the overall training paradigm for MOSE (see Algo. 1 in Appendix 8):

$$\mathcal{L}_{\text{MOSE}} = \mathcal{L}_{\text{MLS}} + \mathcal{L}_{\text{RSD}}. \tag{10}$$

MOSE is highly generalized and possible to be assembled with most replay-based methods (replace $\mathcal{L}_{ce}$ and $\mathcal{L}_{scl}$ with others) due to its innovative focus on network structure. We then empirically demonstrate the superiority of MOSE and the effectiveness of its building blocks in addressing OCL.

5. Experiment

5.1. Experimental Setup

Dataset. We examine two benchmark datasets widely used in OCL, *i.e.*, CIFAR-100 (100 classes) [34] and Tiny-ImageNet (200 classes) [36]. Following the setup of previous OCL research [21, 22, 67], we split CIFAR-100 into 10 disjoint tasks with 10 classes per task (500/100 training/testing samples per class) and split Tiny-ImageNet into 100 disjoint tasks with 2 classes per task (500/50 training/testing samples per class), respectively.

Baseline. We compare our approach with 11 OCL methods (AGEM [9], ER [10], MIR [3], GSS [4], ASER [51], ER-AML [7], GDumb [48], SCR [40], OCM [21], OnPro [67] and GSA [22]) and 7 offline CL methods (DER++ [6], IL2A [75], CO^{2}L [8], LUCIR [28], CCIL [45], BIC [69] and SSIL [1]). Those offline CL methods are trained with one epoch in each task following the online setting. We use average accuracy (**ACC**) and average forgetting (**AF**) of all tasks after learning each task as evaluation metrics [20, 51] (detailed in Appendix 8). Results of all methods and their publication dates are recorded in Tab. 2.

Implementation Details. We use ResNet18 [25] with random initialization as the backbone feature extractor f_θ, which is composed of 4 convolutional blocks corresponding to the 4 experts $\{E_i\}_{i=1}^4$. We implement the feature alignment modules $\{a_{\omega_i}\}_{i=1}^4$ as convolutional layers (group convolution layers with kernel size 3×3 to shrink feature map size and point convolutional layers with kernel size 1×1 to increase channel dimension). The final expert E_4 is feature extractor f_θ itself, so a_{ω_4} is the trivial identity function. We use fully-connected layers as projection layers $\{p_{\psi_i}\}_{i=1}^4$ and $\{g_{\phi_i}\}_{i=1}^4$. The original ResNet18 has 11.29M parameters, and our auxiliary networks bring another 1.52M. Still, they are *far less* than methods requiring knowledge distillation with a copy of previous model [21, 37] and will be *deleted* once the training is completed. An NCM classifier [40, 44] is used to do class prediction with the feature maps output by f_θ (*i.e.*, the final expert E_4). MOSE applies reservoir sampling strategy [58] and random memory update. We fix incoming batch $B = 10$ and buffer batch size $B^{\mathcal{M}} = 64$ for all baselines according to [20, 21, 51]. For the two OCL benchmarks, we set memory size $M = 1k, 2k, 5k$ and $M = 2k, 4k, 10k$, respectively (same as work [21, 22]). We take Adam as our optimizer, with learning rate 1×10^{-3} and weight decay 1×10^{-4} for all methods. τ is $\mathcal{L}_{scl}$ is set to 0.07 as SCR. Default hyperparameters, as well as code

Method	Split CIFAR-100 (10 tasks)						Split Tiny-ImageNet (100 tasks)					
Memory Size	$M=1k$		$M=2k$		$M=5k$		$M=2k$		$M=4k$		$M=10k$	
Metric	ACC(%)↑	AF(%)↓	ACC(%)↑	AF(%)↓	ACC(%)↑	AF(%)↓	ACC(%)↑	AF(%)↓	ACC(%)↑	AF(%)↓	ACC(%)↑	AF(%)↓
AGEM(2019)	5.8±0.2	77.6±2.0	5.9±0.3	76.9±1.5	6.1±0.4	78.3±1.2	0.9±0.1	73.9±0.2	2.0±0.5	77.9±0.2	3.9±0.2	74.1±0.3
ER(2019)	15.7±0.3	66.1±1.3	21.3±0.5	59.3±0.9	28.8±0.8	60.0±1.6	4.7±0.5	68.2±2.8	10.1±0.7	66.2±0.8	11.7±0.2	67.2±0.2
MIR(2019)	16.0±0.4	24.5±0.3	19.0±0.1	21.4±0.3	24.1±0.2	21.0±0.1	6.1±0.5	61.1±3.2	11.7±0.2	60.4±0.5	13.5±0.2	59.5±0.3
GSS(2019)	11.1±0.2	73.4±4.2	13.3±0.5	69.3±3.1	17.4±0.1	70.9±2.9	3.3±0.5	72.8±1.2	10.0±0.2	72.6±0.4	10.5±0.2	71.5±0.2
ASER(2021)	16.4±0.3	25.0±0.2	12.2±1.9	12.2±1.9	27.1±0.3	13.2±0.1	5.3±0.3	65.7±0.7	8.2±0.2	64.2±0.2	10.3±0.4	62.2±0.1
ER-AML(2022)	16.1±0.4	51.5±0.8	17.6±0.5	49.2±0.5	22.6±0.1	38.7±0.6	5.4±0.2	47.4±0.5	7.1±0.5	43.2±0.3	10.1±0.4	41.0±0.5
GDumb(2020)	17.1±0.4	16.7±0.5	25.1±0.2	17.6±0.2	38.6±0.5	16.8±0.4	12.6±0.1	15.9±0.5	12.7±0.3	14.6±0.3	15.7±0.2	11.7±0.2
SCR(2021)	27.3±0.4	17.5±0.2	30.8±0.5	11.6±0.5	36.5±0.3	5.6±0.4	12.6±1.1	19.4±0.3	18.2±0.1	15.4±0.3	21.1±1.1	14.9±0.7
OCM(2022)	28.1±0.3	12.2±0.3	35.0±0.4	8.5±0.3	42.4±0.5	4.5±0.3	15.7±0.2	23.5±1.9	21.2±0.4	21.0±0.3	27.0±0.3	18.6±0.5
OnPro(2023)	30.0±0.4	10.4±0.5	35.9±0.6	6.1±0.6	41.3±0.5	5.3±0.6	16.9±0.4	17.4±0.4	22.1±0.4	16.8±0.4	29.8±0.5	14.6±0.3
GSA(2023)	31.4±0.2	33.2±0.6	39.7±0.6	22.8±0.4	49.7±0.2	8.7±0.3	18.4±0.4	35.5±0.3	26.0±0.2	25.8±0.4	33.2±0.4	16.9±0.6
DER++(2020)	15.3±0.2	43.4±0.2	19.7±1.5	44.0±1.9	27.0±0.7	25.8±3.5	4.5±0.3	67.2±1.7	10.1±0.3	63.6±0.3	17.6±0.5	55.2±0.7
IL2A(2021)	18.2±1.2	24.6±0.6	19.7±0.5	12.5±0.7	22.4±0.2	20.0±0.5	5.5±0.7	65.5±0.7	8.1±1.2	60.1±0.5	11.6±0.4	57.6±1.1
Co^{2}L(2021)	17.1±0.4	16.9±0.4	24.2±0.2	16.6±0.6	32.2±0.5	9.9±0.7	10.1±0.2	60.5±0.5	15.8±0.4	52.5±1.2	22.5±1.2	42.5±0.8
LUCIR(2019)	8.6±1.3	60.0±0.1	19.5±0.7	47.5±0.9	16.9±0.5	44.3±0.7	7.6±0.5	46.4±0.7	9.6±0.7	42.2±0.9	12.5±0.7	37.6±0.7
CCIL(2021)	18.5±0.3	16.7±0.5	19.1±0.4	16.1±0.3	20.5±0.3	17.5±0.2	5.6±0.9	59.4±0.3	7.0±0.5	56.2±1.3	15.2±0.5	48.9±0.6
BiC(2019)	21.2±0.3	40.2±0.4	36.1±1.3	30.9±0.7	42.5±1.2	18.7±0.5	10.2±0.9	43.5±0.5	18.9±0.3	32.9±0.5	25.2±0.6	24.9±0.4
SSIL(2021)	26.0±0.1	40.1±0.5	33.1±0.5	33.9±1.2	39.5±0.4	21.7±0.8	9.6±0.7	44.4±0.7	15.2±1.5	36.6±0.7	21.1±0.1	29.0±0.7
MOSE	35.1±0.4	36.9±0.3	45.1±0.3	25.4±0.4	54.8±0.4	13.5±0.5	19.4±0.5	45.7±1.0	28.0±0.8	29.5±0.8	38.7±0.4	15.5±0.3
MOE-MOSE	37.4±0.3	34.7±0.3	47.0±0.4	23.6±0.4	55.6±0.4	12.7±0.4	21.4±0.4	40.6±0.6	29.8±0.5	26.3±0.5	39.3±0.8	13.9±0.6

Table 2. **Average Accuracy (ACC) & Average Forgetting (AF)** across two class-incremental datasets with 3 different memory sizes. We record the mean and standard deviation from 15 random runs. The best and second best results are highlighted using **bold** and *italic* fonts, respectively. The first 11 rows are online CL algorithms, while the next 7 rows show the performance of offline CL methods. Our proposed method MOSE along with its MOE version are listed in the last two rows, achieving state-of-the-art performance.

links of all baselines, can be found in Appendix 10, where their training time is provided to assess their efficiency.

Data Augmentation. As discussed in Sec. 3.3 we see the positive effect of data augmentation in OCL. Following SimCLR [11] and OCM [21], we use a transformation operation combining random horizontal flip, random grayscale, and random resized crop for our MOSE as well as all baselines for fair comparison. Specifically, augment both incoming batch $\mathcal{B}^t$ and buffer batch $\mathcal{B}^{\mathcal{M}}$. Notice that SCR [40], DER++ [6], and DVC [20] have implemented their unique augmentation transformations, so we keep the default setup. Global rotation augmentation with inner flip introduced in OCM produces 15 times more training samples and is accepted by later works (GSA [22], OnPro [67]). Here we take the inner flip operation to double the training samples, whose impact is discussed in Appendix 9.2.

5.2. Experimental Result

Overall Performance. The overall performance of all methods is summarized in Tab. 2 with the mean and standard deviation of 15 runs. It can be seen that the proposed MOSE framework achieves substantial average accuracy improvement over all baselines, with a performance lead of more than 5.4% and 5.5% compared with state-of-the-art methods over Split CIFAR-100 and Split Tiny-ImageNet, respectively. Besides, we provide the results of MOSE without using RSD, instead, we calculate the performance of average output logits from each expert, denoted as

MOE-MOSE. The results of MOE-MOSE further indicate the superiority of utilizing multiple experts' skills, and it achieves even higher improvement: 7.3% on Split CIFAR-100 and 6.1% on Split Tiny-ImageNet. The outstanding performance on the ACC metric highlights the excellence of our approach to addressing the OCL problem. MOSE's average forgetting over benchmarks is a relatively smaller value than most OCL and offline CL problems. Along with the ACC results, we can conclude that our MOSE gets high test accuracy for each new task when it is under training, and the forgetting does not significantly affect the final performance so ultimately MOSE demonstrates the strongest among all baselines.

Number of Experts. The selection $n = 4$ naturally comes from the architecture of the ResNet, as discussed in Sec. 5.1. In Fig. 3, we show ACC and AF for different n and buffer sizes over Split-CIFAR100. $n = 4$ balances both sides, w/o increasing too much complexity or cost.

Small Buffers. For scalability and transferability concerns, we present experiment results with small buffers in Tab. 3. Here we show the ACC for both datasets except we utilize a 10-task configuration for Split Tiny-ImageNet due to the subpar performance of all baselines in the 100-task version when constrained by limited memory capacities, rendering them inadequate for comparison.

Addressing Overfitting-Underfitting. In response to the overfitting-underfitting dilemma discussed in Sec. 3, we record the new task test accuracy and average BOF (Eq.5)

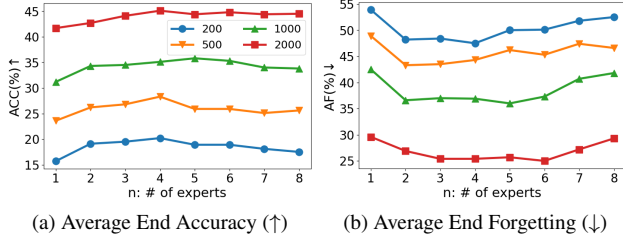(a) Average End Accuracy (↑)   (b) Average End Forgetting (↓)

Figure 3. **Different Number of Experts.** We divide the ResNet18 backbone into a few components according to its block-wise structure, evaluated under four different memory buffer sizes.
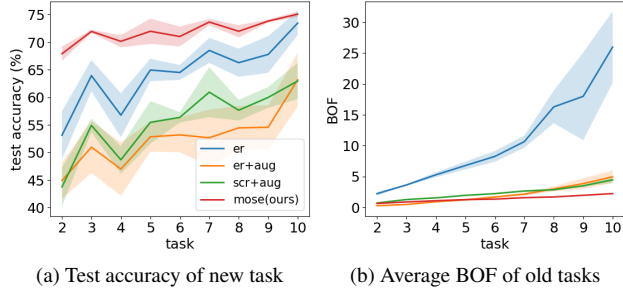

(a) Test accuracy of new task   (b) Average BOF of old tasks

Figure 4. **Overfitting-Underfitting Test**. These two subfigures exhibit **(a)** the test accuracy of each new task t when it is trained; and **(b)** the average BOF value of old tasks after learning each task.

Dataset	Split CIFAR-100 (10 tasks)		Split Tiny-ImageNet(10 tasks)	
Memory Size	200	500	500	1K
OCM(2022)	12.2±0.4	19.7±0.5	7.3±0.5	10.5±0.6
OnPro(2023)	14.1±0.9	21.5±1.4	7.2±0.4	10.2±0.3
GSA(2023)	14.9±0.3	22.9±0.2	10.4±0.3	14.8±0.2
MOSE(ours)	**20.2±0.5**	**28.3±0.7**	**15.2±0.7**	**20.2±0.9**

Table 3. **Small Buffer Scenario.** ACC recorded for 15 runs.

over old tasks in Fig.4, following the same experiment setup in Sec.3.3. These two subfigures demonstrate that MOSE learns each new task well with higher new task test accuracy (avoid underfitting of new task) and mitigates buffer overfitting for old tasks (lower BOF value throughout the entire training period), therefore it resolves both sides of this dilemma pretty well, which explains the origin of its advantages in OCL benchmarks.

5.3. Ablation Study

For each building block of our design, multi-level experts with supervision $\mathcal{L}_{MLS}$ and reverse self-distillation $\mathcal{L}_{RSD}$, we experiment on dataset Split CIFAR-100 with memory size $M = 5k$ (see Tab. 4). We observe a consistent increase in average accuracy with the inclusion of each component. This validates the efficacy of individual modules and their collaborative contribution to the success of MOSE.

Also, Tab. 1 verifies the effectiveness of RSD: the mission of it – concentrates experts' capabilities onto the final

Loss	$\mathcal{L}_{ce}$	$\mathcal{L}_{scl}$	$\mathcal{L}_{RSD}$	**ACC (%)**
w/o MLS	last layer			36.89
		last layer		39.36
	last layer	last layer		45.00
w/ MLS	✓			40.24
	✓	✓		46.21
	✓	✓		52.29
	✓	✓	✓	**54.57**

Table 4. **Ablation Study** with impact of individual building bricks of MOSE over Split CIFAR-100 dataset and $M = 5000$. Records are averaged over 15 random runs. The best result is in **bold** font.

expert – is accomplished. Simultaneously, with the network divided into multiple experts, they each excel in different tasks, forming a *diagonal pattern* where the maximum values for each task are presented in the first half of Tab. 1. On the one hand, this indicates differences in the learned encoding abilities between network hierarchies, with shallower layers possibly possessing better generalization, preserving better accuracy on the initial task by avoiding drastic feature changes during subsequent training. On the other hand, the features obtained in the final layer may be more advanced and semantic, leading to good performance on newer tasks. RSD helps the final expert E_4 learn tasks in which it performs poorly, showing the efficacy of knowledge transfer within the learner network in OCL.

6. Conclusion and Discussion

This paper attributes the distinction between OCL and regular CL to the particular challenge termed overfitting-underfitting dilemma for sequentially arriving tasks. The proposed method MOSE utilizes multi-level latent experts and self-distillation to integrate different skills across network hierarchies, achieving appropriate convergence in each task. Extensive experiments demonstrate the remarkable superiority of MOSE over state-of-the-art baselines and its effectiveness in addressing the overfitting-underfitting dilemma. We expect this to facilitate the understanding of OCL and inspire subsequent research of latent expertise for CL-related problems. In the future, we will combine MOSE with more backbone architectures and different supervision signals to further unleash its potential.

7. Acknowledgement

References

[1] 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 824–833, 2021. 5, 6

[2] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7120–7129, 2017. 5

[3] Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In *Advances in Neural Information Processing Systems*, pages 11849–11860, 2019. 2, 3, 6

[4] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *Advances in Neural Information Processing Systems*, pages 11816–11825, 2019. 3, 6

[5] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems*, pages 2654–2662, 2014. 2

[6] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *Advances in Neural Information Processing Systems*, 2020. 2, 6, 7

[7] Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. In *International Conference on Learning Representations*, 2022. 2, 5, 6

[8] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co^{2}l: Contrastive continual learning. In *2021 IEEE/CVF International Conference on Computer Vision*, pages 9496–9505, 2021. 2, 6

[9] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *International Conference on Learning Representations*, 2019. 2, 6

[10] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet Kumar Dokania, Philip H. S. Torr, and Marc'Aurelio Ranzato. Continual learning with tiny episodic memories, 2019. 2, 3, 6

[11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607, 2020. 4, 7, 2

[12] Haoyang Cheng, Haitao Wen, Xiaoliang Zhang, Heqian Qiu, Lanxiao Wang, and Hongliang Li. Contrastive continuity on augmentation stability rehearsal for continual self-supervised learning. In *IEEE/CVF International Conference on Computer Vision*, pages 5707–5717, 2023. 1

[13] Aristotelis Chrysakis and Marie-Francine Moens. Online continual learning from imbalanced data. In *International Conference on Machine Learning*, pages 1952–1961, 2020. 3

[14] Lee Cossell, Maria Florencia Iacaruso, Dylan R Muir, Rachael Houlton, Elie N Sader, Ho Ko, Sonja B Hofer, and Thomas D Mrsic-Flogel. Functional organization of excitatory synaptic strength in primary visual cortex. *Nature*, 518 (7539):399–403, 2015. 1

[15] Daniel Deitch, Alon Rubin, and Yaniv Ziv. Representational drift in the mouse visual cortex. *Current Biology*, 31(19): 4327–4339, 2021. 4

[16] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, 1999. 2

[17] Qiang Gao, Xiaojun Shan, Yuchen Zhang, and Fan Zhou. Enhancing knowledge transfer for task incremental learning with data-free subnetwork. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[18] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013. 2

[19] Kalanit Grill-Spector and Rafael Malach. The human visual cortex. *Annu. Rev. Neurosci.*, 27:649–677, 2004. 1, 4

[20] 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 7432–7441, 2022. 3, 6, 7, 1

[21] Yiduo Guo, Bing Liu, and Dongyan Zhao. Online continual learning through mutual information maximization. In *International Conference on Machine Learning*, pages 8109–8126, 2022. 2, 4, 6, 7, 1

[22] Yiduo Guo, Bing Liu, and Dongyan Zhao. Dealing with cross-task class discrimination in online continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11878–11887, 2023. 2, 5, 6, 7

[23] Jiangpeng He and Fengqing Zhu. Online continual learning for visual food classification. In *IEEE/CVF International Conference on Computer Vision Workshops*, pages 2337–2346, 2021. 3

[24] Jiangpeng He, Runyu Mao, Zeman Shao, and Fengqing Zhu. Incremental learning in online scenario. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2

[25] 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. 3, 6, 1

[26] Xu He and Herbert Jaeger. Overcoming catastrophic interference using conceptor-aided backpropagation. In *International Conference on Learning Representations*, 2018. 2

[27] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015. 2

[28] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 831–839, 2019. 6

[29] Alumit Ishai, Leslie G Ungerleider, Alex Martin, Jennifer L Schouten, and James V Haxby. Distributed representation of objects in the human ventral visual pathway. *Proceedings of*

the National Academy of Sciences, 96(16):9379–9384, 1999. 1, 4

[30] Xisen Jin, Arka Sadhu, Junyi Du, and Xiang Ren. Gradient-based editing of memory examples for online task-free continual learning. In *Advances in Neural Information Processing Systems*, pages 29193–29205, 2021. 3

[31] Dahuin Jung, Dongjin Lee, Sunwon Hong, Hyemi Jang, Ho Bae, and Sungroh Yoon. New insights for the stability-plasticity dilemma in online continual learning. In *International Conference on Learning Representations*, 2023. 1, 2, 4

[32] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. 2

[33] Dwight J Kravitz, Kadharbatcha S Saleem, Chris I Baker, Leslie G Ungerleider, and Mortimer Mishkin. The ventral visual pathway: an expanded neural framework for the processing of object quality. *Trends in Cognitive Sciences*, 17 (1):26–49, 2013. 1, 4

[34] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 3, 4, 6

[35] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory G. Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(7):3366–3385, 2022. 1, 2

[36] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 6

[37] Zhizhong Li and Derek Hoiem. Learning without forgetting. In *European Conference on Computer Vision*, pages 614–629, 2016. 2, 6

[38] Huiwei Lin, Baoquan Zhang, Shanshan Feng, Xutao Li, and Yunming Ye. PCR: proxy-based contrastive replay for online class-incremental continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24246–24255, 2023. 1, 3

[39] Yilin Lyu, Liyuan Wang, Xingxing Zhang, Zicheng Sun, Hang Su, Jun Zhu, and Liping Jing. Overcoming recency bias of normalization statistics in continual learning: Balance and adaptation. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[40] 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 Conference on Computer Vision and Pattern Recognition Workshops,*, pages 3589–3599, 2021. 1, 2, 3, 6, 7

[41] 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, 2

[42] Marc Masana, Xialei Liu, Bartlomiej Twardowski, Mikel Menta, Andrew D. Bagdanov, and Joost van de Weijer. Class-incremental learning: Survey and performance evaluation on image classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(5):5513–5533, 2023. 1, 2

[43] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, pages 109–165. Elsevier, 1989. 2

[44] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(11):2624–2637, 2013. 3, 6, 1

[45] Sudhanshu Mittal, Silvio Galesso, and Thomas Brox. Essentials for class incremental learning. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 3513–3522, 2021. 6

[46] Oleksiy Ostapenko, Mihai Marian Puscas, Tassilo Klein, Patrick Jähnichen, and Moin Nabi. Learning to remember: A synaptic plasticity driven framework for continual learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 11321–11329, 2019. 2

[47] German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019. 1, 2

[48] Ameya Prabhu, Philip H. S. 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, 2020. 2, 6

[49] 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 5533–5542, 2017. 2

[50] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning*, pages 4548–4557, 2018. 2

[51] Dongsub Shim, Zheda Mai, Jihwan Jeong, Scott Sanner, Hyunwoo Kim, and Jongseong Jang. Online class-incremental continual learning with adversarial shapley value. In *AAAI Conference on Artificial Intelligence*, pages 9630–9638, 2021. 2, 3, 6, 1

[52] Albin Soutif-Cormerais, Antonio Carta, Andrea Cossu, Julio Hurtado, Vincenzo Lomonaco, Joost Van de Weijer, and Hamed Hemati. A comprehensive empirical evaluation on online continual learning. In *IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 3518–3528, 2023. 1, 2

[53] Mototaka Suzuki, Cyriel MA Pennartz, and Jaan Aru. How deep is the brain? the shallow brain hypothesis. *Nature Reviews Neuroscience*, 24(12):778–791, 2023. 1

[54] Peter T. Szymanski and Michael D. Lemmon. Adaptive mixtures of local experts are source coding solutions. In *International Conference on Neural Networks*, pages 1391–1396, 1993. 5

[55] Frank Tong. Primary visual cortex and visual awareness. *Nature Reviews Neuroscience*, 4(3):219–229, 2003. 1, 4

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. 3

[57] Eli Verwimp, Matthias De Lange, and Tinne Tuytelaars. Rehearsal revealed: The limits and merits of revisiting samples in continual learning. In *IEEE/CVF International Conference on Computer Vision*, pages 9365–9374, 2021. 1

[58] Jeffrey Scott Vitter. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11(1):37–57, 1985. 2, 3, 6

[59] Johannes von Oswald, Christian Henning, João Sacramento, and Benjamin F. Grewe. Continual learning with hypernetworks. In *International Conference on Learning Representations*, 2020. 2

[60] Liyuan Wang, Kuo Yang, Chongxuan Li, Lanqing Hong, Zhenguo Li, and Jun Zhu. Ordisco: Effective and efficient usage of incremental unlabeled data for semi-supervised continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5383–5392, 2021. 2

[61] Liyuan Wang, Mingtian Zhang, Zhongfan Jia, Qian Li, Chenglong Bao, Kaisheng Ma, Jun Zhu, and Yi Zhong. Afec: Active forgetting of negative transfer in continual learning. *Advances in Neural Information Processing Systems*, 34: 22379–22391, 2021. 2

[62] Liyuan Wang, Xingxing Zhang, Qian Li, Jun Zhu, and Yi Zhong. Coscl: Cooperation of small continual learners is stronger than a big one. In *European Conference on Computer Vision*, pages 254–271, 2022. 2, 5

[63] Liyuan Wang, Xingxing Zhang, Kuo Yang, Longhui Yu, Chongxuan Li, Lanqing Hong, Shifeng Zhang, Zhenguo Li, Yi Zhong, and Jun Zhu. Memory replay with data compression for continual learning. In *International Conference on Learning Representations*, 2022. 2

[64] Liyuan Wang, Xingxing Zhang, Qian Li, Mingtian Zhang, Hang Su, Jun Zhu, and Yi Zhong. Incorporating neuro-inspired adaptability for continual learning in artificial intelligence. *Nature Machine Intelligence*, 5(12):1356–1368, 2023. 2

[65] Liyuan Wang, Jingyi Xie, Xingxing Zhang, Mingyi Huang, Hang Su, and Jun Zhu. Hierarchical decomposition of prompt-based continual learning: Rethinking obscured suboptimality. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[66] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2024. 1, 2

[67] Yujie Wei, Jiaxin Ye, Zhizhong Huang, Junping Zhang, and Hongming Shan. Online prototype learning for online continual learning. In *IEEE/CVF International Conference on Computer Vision*, pages 18764–18774, 2023. 2, 6, 7

[68] Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in superposition. In *Advances in Neural Information Processing Systems*, 2020. 2

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

[70] Guangyu Robert Yang and Xiao-Jing Wang. Artificial neural networks for neuroscientists: A primer. *Neuron*, 107(6): 1048–1070, 2020. 4

[71] Da Yu, Mingyi Zhang, Mantian Li, Fusheng Zha, Junge Zhang, Lining Sun, and Kaiqi Huang. Contrastive correlation preserving replay for online continual learning. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2023. 1, 2, 3

[72] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *IEEE/CVF International Conference on Computer Vision*, pages 3712–3721, 2019. 4, 5, 2

[73] Yaqian Zhang, Bernhard Pfahringer, Eibe Frank, Albert Bifet, Nick Jin Sean Lim, and Yunzhe Jia. A simple but strong baseline for online continual learning: Repeated augmented rehearsal. In *Advances in Neural Information Processing Systems*, 2022. 1, 3

[74] Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Deep class-incremental learning: A survey. *CoRR*, abs/2302.03648, 2023. 1, 2

[75] Fei Zhu, Zhen Cheng, Xu-Yao Zhang, and Chenglin Liu. Class-incremental learning via dual augmentation. In *Advances in Neural Information Processing Systems*, pages 14306–14318, 2021. 1, 6