Enhancing Class-Incremental Learning for Image Classification via Bidirectional Transport and Selective Momentum

Feifei Fu
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
fufeifei@ruc.edu.cn

Yizhao Gao
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
gaoyizhao@ruc.edu.cn

Zhiwu Lu
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
luzhiwu@ruc.edu.cn

ABSTRACT

Class-Incremental Learning (Class-IL) aims to continuously learn new knowledge without forgetting old knowledge from a given data stream in the realm of image classification. Recent Class-IL methods strive to balance old and new knowledge and have achieved excellent results in mitigating the forgetting by mainly employing the rehearsal-based strategy. However, the representation learning on new tasks is often impaired since the trade-off is hard to taken between old and new knowledge. To overcome this challenge, based on the Complementary Learning System (CLS) theory, we propose a novel CLS-based method by focusing on the representation of old and new knowledge under the Class-IL setting, which can acquire more new knowledge from new tasks while consolidating the old knowledge so as to make a better balance between them (i.e., enhancing the overall model performance). Specifically, our proposed method has two novel components: (1) To effectively mitigate the forgetting, we first propose a bidirectional transport (BDT) strategy between old and new models, which can better integrate the old knowledge into the new knowledge and meanwhile enforce the old knowledge to be better consolidated by bidirectionally transferring parameters across old and new models. (2) To ensure that the representation of new knowledge is not impaired by the old knowledge, we further devise a selective momentum (SMT) mechanism to give parameters greater flexibility to learn new knowledge while transferring important old knowledge, which is achieved by selectively (momentum) updating network parameters through parameter importance evaluation. Extensive experiments on five benchmarks show that our proposed method significantly outperforms the state-of-the-arts under the Class-IL setting.

CCS CONCEPTS

• **Computing methodologies** → **Lifelong machine learning**.

KEYWORDS

Class-incremental learning; continual learning; catastrophic forgetting

ICMR '24, June 10–14, 2024, Phuket, Thailand.

ACM ISBN 979-8-4007-0619-6/24/06
https://doi.org/10.1145/3652583.3658063

ACM Reference Format:
Feifei Fu, Yizhao Gao, and Zhiwu Lu. 2024. Enhancing Class-Incremental Learning for Image Classification via Bidirectional Transport and Selective Momentum. In *Proceedings of the 2024 International Conference on Multimedia Retrieval (ICMR '24), June 10–14, 2024, Phuket, Thailand*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3652583.3658063

1 INTRODUCTION

Catastrophic forgetting [23], i.e., learning new knowledge while forgetting previously learned old knowledge, is a long-standing problem for continual learning. The Class-Incremental Learning (Class-IL) setting, as the setting closest to real-world application scenarios in continual learning, where the catastrophic forgetting problem for image classification has been widely studied through various strategies such as regularization-based strategy [1, 16, 20, 40], architecture-based strategy [22, 30, 34] and rehearsal-based strategy [4, 7, 8, 15, 21, 39, 41, 42] with great success.

Recent methods, striving to balance old and new knowledge while mitigating the catastrophic forgetting, have achieved substantial advancement through the rehearsal-based strategy under the Class-IL setting. There are mainly two lines of works. A line of works [5, 6, 27] resort to simple knowledge distillation [13, 31], i.e., they preserve old knowledge by aligning the output logits of current new model (student model) with the output logits of previous models (teacher model) and adjust loss weights to balance old and new knowledge. Inspired by the Complementary Learning System (CLS) theory [18, 35], another line of works [2, 24, 25, 32, 33] combine the distillation with CLS theory: a short-term model is built for fast learning the episodic knowledge and a long-term model is built for slow learning the general structured knowledge by simulating the two learning systems 'hippocampus' and 'neocortex' in the brain, so that the old and new knowledge can be better obtained. Although great progress has been achieved in balancing old and new knowledge by adjusting loss weights or proposing effective strategies, it is difficult to trade-off between old and new knowledge, resulting in these methods not being stable enough to represent both old and new knowledge well. Particularly, the representation of new knowledge is often impaired (see Figure 1).

To address this problem, we propose a novel CLS-based method by focusing on the representation of old and new knowledge under the Class-IL setting, termed BDT-SMT, which can acquire more new knowledge from new tasks while consolidating the old knowledge so as to make a better balance between them (i.e., enhancing the overall model performance). Specifically, similar to [2], our proposed method (see Figure 2) has three main modules: a working model and two semantic memory models (i.e., the long-term

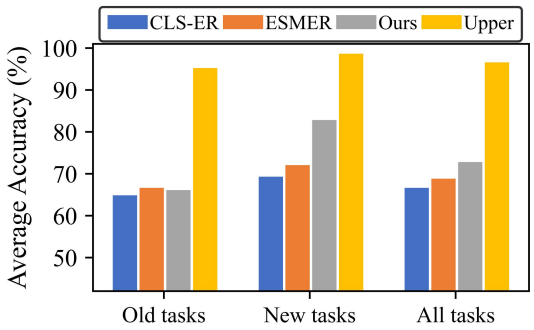

Figure 1: The average accuracy of recent methods (CLS-ER [2], ESMER [32]), our BDT-SMT and 'Upper' on the S-CIFAR-10 (all five tasks) respectively. Here 'Upper' refers to all tasks being sequentially trained with fine-tuning. The accuracy of 'Upper' on each task is obtained by each current model. The accuracy of other methods on each task is obtained by the final model. We define the T_1 to T_3 as old tasks, and the T_4 to T_5 as new tasks. It shows that the accuracy of recent methods suffers more on new tasks compared to ours.

and short-term models that simulate the two learning systems of the brain). Built on this framework, we first devise a bidirectional transport (BDT) strategy to transfer parameters *directly and bidirectionally* between the working model and the two semantic memory models. We denote the direct transport process of parameters (working model → semantic memory models) as backward transport and that (semantic memory models → working model) as forward transport, respectively. This is quite different from [2, 32, 33] with only one unidirectional process (i.e., backward transport). With the BDT strategy, our proposed method forms a circular transport channel among these three models to transfer information to each other more smoothly, thus effectively mitigate the forgetting. Note that the extension to bidirectional transport is *not that easy*: (1) Only one unidirectional process (direct transport of parameters between models) is concerned even in the latest works [32, 33]. (2) Within the CLS framework, the bidirectional process becomes challenging across three models. (3) The forward transport of bidirectional process may impair the representation of new knowledge, which is also the reason why the selective momentum (SMT) mechanism is carefully designed along with BDT in this paper.

Furthermore, to ensure that the representation of new knowledge is not impaired by the (integrated) old knowledge, we devise a selective momentum (SMT) mechanism to selectively (momentum) update parameters with the evaluation of parameter importance during the forward transport. Concretely, a parameter importance evaluation algorithm like SI [40] is introduced into SMT, and an importance threshold is set to control the momentum updates of the parameters, so as to receive more important knowledge from old tasks for important parameters while giving greater flexibility to other unimportant parameters for better learning new tasks. Thus, the designed SMT mechanism in our method is able to enforce the model to continuously transfer important old knowledge as well as learn new knowledge significantly better through selective momentum updating of network parameters. Note that the biggest parameter-updating difference between SI (or EWC [16]) and our BDT-SMT lies in that the parameters of our method are *selectively momentum* updated by the parameters of old model according to parameters importance, while all parameters of SI/EWC are gradient updated by backpropagation according to parameters importance. Additionally, we choose to devise the importance evaluation algorithm according to SI, instead of other strategies [16], because it is more complementary to our proposed method, which has been demonstrated in Section 4.4 (see Table 6).

Our main contributions are four-fold: **(1)** We propose a novel CLS-based method termed BDT-SMT to acquire more new knowledge from new tasks while consolidating the old knowledge so as to make a better balance between them under the Class-IL setting. **(2)** To effectively mitigate the forgetting, we devise a bidirectional transport (BDT) strategy between old and new models, which is quite different from the latest works [2, 32, 33] with only one unidirectional process (i.e., backward transport). Moreover, to ensure that the representation of new knowledge is not impaired by the old knowledge during forward transport, we design a selective momentum (SMT) mechanism to selectively (momentum) update network parameters through parameter importance evaluation. **(3)** Extensive experiments on five benchmarks show that our proposed method significantly outperforms the state-of-the-art methods under the Class-IL setting.

2 RELATED WORK

Knowledge Distillation Knowledge distillation [13] is essential in the rehearsal-based methods [6, 7, 14, 26, 28, 42], which mainly consolidates the old knowledge and mitigates forgetting by distilling knowledge on the replayed data. For example, earlier work iCaRL [27] mitigates forgetting by replaying the exemplars for nearest-mean-of-exemplars classification and aligning the output scores of the previous step with the current step for distillation. Recent work Dark Experience Replay (DER) [6] mitigates forgetting by matching the network's logits of samples at the current step with the logits of previous steps, where samples are replayed from the memory buffer. Most recently, as a modified version of DER, X-DER [5] is proposed, the difference is that X-DER includes a regularly update memory buffer by inserting secondary information ('future past') and a future preparation to incoming tasks. Although X-DER improves the model performance compared to DER, it suffers from a very long training time, hindering its practical application for other datasets.

CLS Theory The CLS theory [18, 35] posits the existence of two interacting systems in the brain: a fast learning system 'hippocampus' and a slow learning system 'neocortex', which prevent forgetting by continuously transferring information. Inspired by this, a line of methods [2, 24, 25, 32, 33] are proposed to mitigate forgetting with a combination of distillation and CLS theory. For example, DualNet [24] is proposed to mitigate forgetting by using a supervised network as fast net for supervised learning and an unsupervised network as slow net for self-supervised learning with the samples replayed from memory buffer. The recent work CLS-ER [2] builds long-term and short-term semantic memory models, and employs the experience replay strategy to align the output logits between the old and new models. SCoMMER [33] proposes a semantic dropout mechanism that simulates the sparse coding idea of the brain, enforcing the model to have similar activation units for semantically similar inputs while reducing overlap for semantically

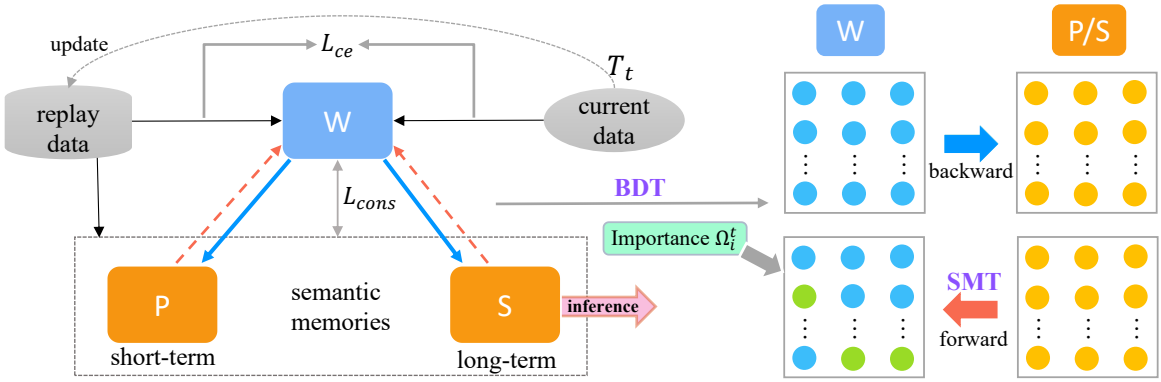

Figure 2: The overview of our BDT-SMT built over the working model (W) and two semantic memory models (plastic model (P) and stable model (S)). The combined use of BDT and SMT strategies forces our model to achieve a better balance between old and new knowledge.

dissimilar inputs. ESMER [32] simulates the brain's idea of learning more information from small errors, and proposes a modulation mechanism based on error sensitivity to help the model learn more information. These methods have achieved promising results in balancing old and new knowledge, but they are not stable enough to represent both the old and new knowledge well such as the representation of new knowledge is often impaired. To address this problem, following [2], we propose a novel and effective method with negligible computation cost (see Table 4).

3 METHODOLOGY

3.1 Problem Definition

The Class-IL setting is concerned for continual learning, where the model is trained for image classification on a sequential tasks T, i.e., $T = \{T_1, T_2, ..., T_H\}$, where H denotes the number of tasks. For each task T_t $(1 \le t \le H)$ from T, it owns a task-specific training set $D_t = \{(x_i^t, y_i^t)\}_{i=1}^{N_t}$ with N_t sample pairs, where $x_i^t \in R^D$ is a sample image from the class $y_i^t \in Y^t$. Y^t is the label space of the task T_t. For label spaces of different tasks T_t and $T_{t'}$, there is non-overlapping classes, i.e., $Y^t \cap Y^{t'} = \emptyset$ for $t \ne t'$. Further, for each task T_t, its validation and test sets can be defined similarly. The goal of Class-IL is to enforce the trained model to accurately classify all previously seen classes with less forgetting after learning all tasks without providing the task identifiers.

3.2 Base Framework

Similar to the recent work [2], the base framework of our BDT-SMT has three main modules: a working model and two semantic memory models (i.e., plastic model and stable model, see Figure 2). These three models are deployed with the same network structure (e.g., ResNet18 [12]) and initialized with the same parameters. The main functions of these modules are described separately below.

Working Model The function of working model is two-fold: on one hand, it is trained to continuously learn the incoming tasks on a given data stream (D) to acquire new knowledge; on the other hand, it needs to continuously transfer the learned knowledge to the semantic memory models to accumulate the old knowledge. Concretely, given the current batch data (X_D, Y_D) from the dataset D_t and the replayed batch data (X_M, Y_M) randomly sampled from the memory buffer M_t for the task T_t as the inputs, the working model $F(;\theta_W)$ is trained to accurately classify the current task data and replayed old data by backpropagation (see Eq. (6) for the total loss). Meanwhile, the working model continuously transfers the learned knowledge (information) stored in θ_W to the two semantic memory models, i.e., it uses the parameters θ_W to update the parameters of semantic memory models, so that the semantic memory models can obtain the long-term and short-term memory retention of knowledge (see below for the detailed transfer process). Thus, the working model plays an important role (updating & transferring parameters) in the framework.

Semantic Memory Models The semantic memory models, i.e., stable model (S) and plastic model (P), are maintained to retain the knowledge previously learned on the working model. The stable model $F(;\theta_S)$ continuously accumulates the knowledge as long-term memory to acquire the slow adaptation of information, while the plastic model $F(;\theta_P)$ continuously accumulates the knowledge as short-term memory to acquire the fast adaptation of information. The realization of long-term and short-term period mainly depends on the update frequency of parameters. Thus, instead of updating parameters at each training step, the frequency parameters are employed on semantic memory models to stochastically control their updates. Concretely, the long-term stable model is used to retain more information of earlier tasks with a small frequency parameter f_S, while the short-term plastic model is used to retain more information of recent tasks with a big frequency parameter f_P $(f_P \ge f_S)$. Compared with the short-term memory model, the long-term memory model can accumulate more general structured knowledge over time, leading to better generalization across tasks. As a result, the stable model is used for inference.

Given the parameters of working model θ_W, the parameters of plastic model θ_P and stable model θ_S, the parameters of two semantic memory models θ_P, θ_S are (momentum) updated by the parameters of working model θ_W through the Exponential Moving Average (EMA) strategy [11, 37], which are formulated as:

$$\begin{aligned} \theta_P &= m_P \cdot \theta_P + (1 - m_P) \cdot \theta_W, \text{ if rand(1)} < f_P \\ \theta_S &= m_S \cdot \theta_S + (1 - m_S) \cdot \theta_W, \text{ if rand(1)} < f_S \end{aligned}, \tag{1}$$

where m_P and $m_S \in [0, 1)$ denote the backward transport momentum parameters, and rand(1) denotes a random value sampled from a standard Gaussian distribution. By setting $m_P \le m_S$, the plastic

model can adjust new information faster (more), while the stable model can adjust new information slower (less) so as to construct the general structured knowledge over time. Note that the parameters of working model θ_W are updated by backpropagation, and the parameters of two semantic memory models are only updated by θ_W since they have no gradients.

3.3 Our BDT-SMT Method

In this paper, built on the above base framework, our BDT-SMT devises two novel components, i.e., bidirectional transport (BDT) strategy and selective momentum (SMT) mechanism. The combination of BDT and SMT facilitates the model to acquire more new knowledge from new tasks while consolidating the old knowledge, thereby achieving a better balance between old and new knowledge.

Bidirectional Transport (BDT) Strategy To effectively mitigate the forgetting, we propose a BDT strategy (including forward & backward transport) to transfer information more smoothly between the three models in the form of a circular transport channel ($P \Leftrightarrow W \Leftrightarrow S$), as shown in Figure 2. Compared to the unidirectional transport (only backward transport) used in [2], the BDT strategy indeed strengthens the information communication among the three models: it enables the knowledge of earlier tasks (from stable model), the knowledge of recent tasks (from plastic model) and the knowledge of new tasks (from working model) to be continuously transferred and flowed between models, thereby better consolidating the old knowledge and mitigating the forgetting. Concretely, at each training step, the parameters of working model (θ_W) are first updated by backpropagation. Then, with the updated working model, it is required to determine whether to momentum update the plastic model and stable model according to Eq. (1) (i.e., backward transport). Let U_{PS} denote the set of semantic memory models that are updated by the working model (U_{PS} may be $\emptyset$, $\{P\}$, $\{S\}$, and $\{P, S\}$). In turn, the updated semantic memory models U_{PS} are used to momentum update the working model at the beginning of the next training step (i.e., forward transport). Formally, given the parameters of the three models θ_W, θ_P and θ_S, the momentum update of the parameters of the working model in forward transport adopts the same EMA as in Eq. (1) with the forward transport momentum parameter $\hat{m} \in [0, 1)$:

$$\theta_W = \hat{m} \cdot \theta_W + (1 - \hat{m}) \cdot \theta_j, \ j \in U_{PS}. \tag{2}$$

In our experiments, we find that the plastic model must be used for momentum updating before the stable model when $U_{PS} = \{P, S\}$. And note that the forward transport begins with the second task to ensure that semantic memory models have accumulated knowledge.

Selective Momentum (SMT) Mechanism Although good retention of old knowledge and promising improvement of model performance are obtained with the BDT strategy (see Table 3), there still exists a clear limitation in the representation of new knowledge. Specifically, in the forward transport of BDT, all parameters of the working model are momentum updated by the parameters of the semantic memory models. Among these parameters of semantic memory models, some parameters are helpful (positive) to the representation of new knowledge, such as parameters representing general structured knowledge; while some parameters are useless or even harmful (negative) to the representation of new knowledge, resulting in that the new knowledge is not well represented. Thus, here we design a SMT mechanism to ensure that the representation of new knowledge is not impaired by the integrated old knowledge, which is achieved by selectively updating parameters with the evaluation of parameters importance during forward transport.

The SMT mechanism is designed to give greater flexibility to unimportant parameters of previous tasks for better learning new tasks. Concretely, we first need to evaluate the importance of parameters of working model by the importance evaluation algorithm, then rank the parameter importance and set an importance threshold k (k is a fraction) to indicate the parameter range that needs to be updated. As a result, the Top-k part of parameters in the working model that are important to previous tasks are momentum updated by the corresponding Top-k part of parameters of semantic memory models, and other unimportant parameters of working model are no longer updated by the semantic memory models (see Figure 5 for the selection of k). Let V_{top} denote the set of important Top-k part of parameters. The forward transport is reformulated as:

$$\theta_W[i] = \hat{m} \cdot \theta_W[i] + (1 - \hat{m}) \cdot \theta_j[i], \ j \in U_{PS}, \ i \in V_{top}, \tag{3}$$

where $\theta_W[i]$ (or $\theta_j[i]$) is the i-th element of θ_W (or θ_j). With this mechanism, our BDT-SMT can receive more important knowledge from old tasks for important parameters while giving greater flexibility to unimportant parameters for fitting new tasks.

For the importance evaluation algorithm, it is devised according to SI [40]. Differently, in SI, the parameters importance is used as a penalty strength in the loss function to enforce all parameters to be gradient updated by backpropagation; in our BDT-SMT, the parameters importance is used as a criterion to determine whether the parameter needs to be updated, so that the parameters can be *selectively momentum* updated according to the threshold k. Formally, when the importance of parameter $\theta_W[i]$ for task t (T_t) is denoted as ω_i^t, we can approximately define it as the contribution of $\theta_W[i]$ to the change of the total loss throughout the training phase of task t:

$$\omega_i^t \equiv \sum_s \eta \cdot g_{s,t}^2(\theta_W[i]), \tag{4}$$

where η is the learning rate, and $g_{s,t}(\theta_W[i])$ is the gradient of $\theta_W[i]$ at the training step s for task t. Notably ω_i^t only measures the absolute contribution of the parameter, ignoring its own update range. Thus, a normalized importance (regularization strength) Ω_i^t for task t is given by:

$$\Omega_i^t = \sum_{\tau \le t} \frac{\omega_i^\tau}{(\triangle_i^\tau)^2 + \varepsilon}, \ \triangle_i^\tau \equiv \theta_W^\tau[i] - \theta_W^{\tau-1}[i], \tag{5}$$

where task τ denotes a task before task t. $\triangle_i^\tau$ denotes the update amount of $\theta_W[i]$ in task τ (θ_W^τ denotes θ_W at the end of task τ). ε is a small constant used to prevent calculation instability. Note that ω_i^t is initialized to zero at the beginning of each task, while Ω_i^t is only initialized to zero at the beginning of the first task and updated by the accumulated ω_i^t at the end of each task. The computation cost is thus negligible compared to [2] (see Table 4 for a detailed comparative analysis). Additionally, the pseudocode of importance evaluation algorithm is provided in Alg. 1.

By deploying SMT to selectively update the parameters of working model, the transfer of harmful old parameters is effectively avoided, enforcing the working model to learn richer representation of new knowledge. As a result, under the joint action of our

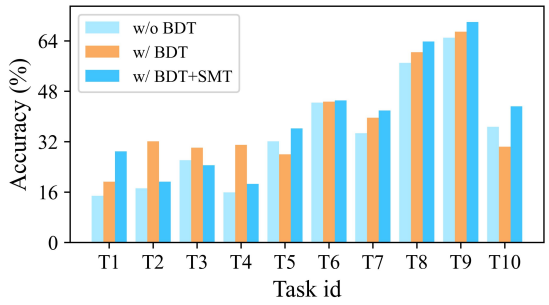

Figure 3: Comparative results of the final model on each learned task of S-CIFAR-100 among the base framework without BDT ("w/o BDT"), with BDT ("w/ BDT"), and with BDT+SMT ("w/ BDT+SMT").

BDT and SMT, the stable model is able to obtain more general structured knowledge over time, which facilitates better generalization across tasks (i.e., the old and new tasks), resulting in superior performance. To make this clearer, in Figure 3 we show the accuracy results of the final model (obtained from one independent run) on ten learned tasks of S-CIFAR-100 among the base framework without BDT, with BDT, and with BDT+SMT. We can see that: (1) Compared with "w/o BDT", the utilization of BDT enforces the model to better consolidate old knowledge on earlier tasks and meanwhile achieve better results on recent tasks. This finding highlights that the knowledge transfer among three models plays a crucial role in facilitating the smoothing of the decision boundaries of new and old tasks (i.e., greatly reducing the distribution overlap between old and new classes), thereby effectively mitigating the forgetting. Meanwhile, the knowledge transfer also brings benefits to new tasks, further demonstrating its importance in continual learning. (2) The utilization of SMT empowers the model to achieve higher accuracies on recent and new tasks. This directly shows that the model's representation ability in new tasks has been substantially improved by applying SMT, thereby facilitating the acquisition of better general structured knowledge over time.

Total Loss The total loss (L) is composed of a cross entropy loss (L_{ce}, standard cross entropy loss) and a consistency loss (L_{cons}). Among them, the cross entropy loss is computed on the current batch data (X_D, Y_D) and replayed batch data (X_M, Y_M), due to its simpleness, we mainly introduce the consistency loss here. Specifically, the consistency loss is computed on the replayed batch data (X_M, Y_M) by aligning the output logits of working model ($Z'_W = F(X_M; \theta_W)$) with the optimal output logits of semantic memory models ($Z_P = F(X_M; \theta_P)$ for plastic model or $Z_S = F(X_M; \theta_S)$ for stable model). The optimal output logits is expressed as a better representation of semantic information for the replay data between plastic model and stable model, i.e., the final selected logits own higher softmax scores for the ground-truth labels of the inputs. Thus, the total loss function L is formulated as:

$$L = L_{ce}(sf(Z_W), Y) + \gamma L_{cons}(Z'_W, Z), \tag{6}$$

$$Z_W = F((X_D \cup X_M); \theta_W), \tag{7}$$

$$L_{cons}(Z'_W, Z) = \frac{1}{BC}\sum_{i=1}^{B}\sum_{j=1}^{C}(Z'_W[i,j] - Z[i,j])^2, \tag{8}$$

where Z_W is the output logits of two batch data X_D and X_M; Z'_W is the output logits of the replayed batch data X_M; Z denotes the optimal output logits of X_M, i.e., Z_S or Z_P. Y refers to the ground-truth labels of X_D and X_M, i.e., $Y = Y_D \cup Y_M$. $\mathrm{sf}(\cdot)$ represents the softmax function, γ represents the balancing hyperparameter. Note that L_{cons} is defined as a Mean Squared Error (MSE) loss. $[B, C]$ refers to the dimension of output logits, and $Z'_W[i,j]$ (or $Z[i,j]$) is the $[i]$-th row and $[j]$-th column element of Z'_W (or Z).

Algorithm 1 Normalized Importance Evaluation

Input: current model $F^t(\cdot)$, current data D_t
current replay data for previous tasks M_t
the normalized importance Ω_i^{t-1} of $\theta_W^{t-1}[i]$ at the end of task $t-1$, learning rate η
Output: the learned normalized importance Ω_i^t
Initialization: $\omega_i^t \leftarrow 0$, $D \leftarrow D_t \cup M_t$, $\theta_W^t[i] \leftarrow \theta_W^{t-1}[i]$
for 1..N epochs **do**
for (x, y) in D **do**
Compute the loss L by Eq. (6);
Compute the gradient:
$g(\theta_W^t[i]) = \nabla_{\theta_W^t[i]} L$ by backpropagation;
Compute the updated parameters:
$\theta_W^t[i] \leftarrow \theta_W^t[i] - \eta \cdot g(\theta_W^t[i])$;
Compute the updated importance:
$\omega_i^t \leftarrow \omega_i^t + \eta \cdot g^2(\theta_W^t[i])$;
end for
end for
Compute the update amount throughout the task t:
$\triangle_i^t \leftarrow \theta_W^t[i] - \theta_W^{t-1}[i]$;
Compute the updated normalized importance:
$\Omega_i^t \leftarrow \Omega_i^{t-1} + \frac{\omega_i^t}{(\triangle_i^t)^2 + \varepsilon}$.
return the updated normalized importance Ω_i^t

4 EXPERIMENTS

4.1 Experimental Setup

Datasets Five standard benchmark datasets are used to evaluate the model performance under the Class-IL setting. (1) **S-MNIST** is obtained by splitting digit-base dataset MNIST [19] into 5 consecutive tasks with two classes per task. For each class in the task, there are 6,000 images for training and 1,000 images for testing. The resolution of these images is $28 * 28$. (2) **S-CIFAR-10** is obtained by splitting the dataset CIFAR-10 [17] into 5 consecutive tasks with two classes per task. For each class in the task, there are 5,000 images for training and 1,000 images for testing. The resolution of these images is $32 * 32 * 3$. (3) **S-CIFAR-100** is obtained by splitting the dataset CIFAR-100 [17] into 10 consecutive tasks with 10 classes per task. For each class in the task, there are 500 images for training and 100 images for testing. These images have the same resolution as S-CIFAR-10. (4) **S-Tiny-ImageNet** is obtained by splitting the dataset Tiny-ImageNet [3] into 10 consecutive tasks with 20 classes per task. For each class in the task, there are 500 images for training and 50 images for testing. The resolution of these images is $64 * 64 * 3$. (5) **S-Mini-ImageNet** is obtained by splitting the dataset miniImageNet [29] (a subset of ImageNet [9]) into 10 tasks with 10 classes per task. For each class in the task, there are 500 images for training and 100 images for testing. The resolution of these images

Table 1: Comparison to the state-of-the-arts under the Class-IL setting in terms of average accuracy over ten independent runs. The standard deviation is given in brackets. All methods (with the same backbone) are trained from scratch.

Method	Buffer Size	S-MNIST	S-CIFAR-10	S-CIFAR-100	S-Tiny-ImageNet	S-Mini-ImageNet
JOINT (upper bound)	–	95.57 (±0.24)	92.20 (±0.15)	70.55 (±0.91)	59.99 (±0.19)	51.40 (±0.31)
SGD (lower bound)	–	19.60 (±0.04)	19.62 (±0.05)	9.32 (±0.06)	7.92 (±0.26)	8.08 (±0.06)
ER [28]	200	80.43 (±1.89)	44.79 (±1.86)	14.78 (±0.67)	8.49 (±0.16)	8.74 (±0.09)
GEM [21]	200	80.11 (±1.54)	25.54 (±0.76)	13.34 (±0.43)	–	–
iCaRL [27]	200	70.51 (±0.53)	49.02 (±3.20)	35.99 (±0.49)	7.53 (±0.79)	20.14 (±0.39)
ER-ACE [7]	200	85.24 (±0.65)	64.08 (±1.68)	27.85 (±0.61)	12.73 (±0.66)	12.94 (±0.43)
DER++ [6]	200	85.61 (±1.40)	64.88 (±1.17)	26.40 (±1.17)	10.96 (±1.17)	12.50 (±0.77)
CLS-ER [2]	200	89.54 (±0.21)	66.19 (±0.75)	35.39 (±1.15)	23.47 (±0.80)	19.41 (±0.91)
SCoMMER [33]	200	–	67.87 (±0.47)	31.75 (±1.39)	16.61 (±0.46)	–
ESMER [32]	200	89.21 (±0.26)	68.51 (±0.33)	35.72 (±0.25)	23.37 (±0.11)	20.46 (±0.40)
BDT-SMT (ours)	200	**89.99 (±0.27)**	**70.19 (±1.13)**	**38.05 (±0.25)**	**25.31 (±0.29)**	**20.82 (±0.60)**

is $84 * 84 * 3$. Note that a fixed order for all classes is kept in each dataset for sequential training across ten independent runs.

Evaluation Metrics To evaluate the model performance in Class-IL, average accuracy (Acc) and average forgetting (Fg) [6, 10, 25] are reported after learning all tasks across ten independent runs. For the accuracy, it refers to the accuracy of the last task on the previous tasks, where a larger value indicates a better model performance (main metric for continual learning). For the forgetting, it refers to the difference between the accuracy of the last task on previous task and the acquired maximum accuracy on this task, where a smaller value indicates a better model performance. Formally, given the number of tasks H and the accuracy $a_{i,j}(j \leq i)$ of the task i on the previous task j, as in previous works, average accuracy Acc (↑) and average forgetting Fg (↓) can be formulated as follows:

$$Acc(\uparrow) = \frac{1}{H}\sum_{j=1}^{H} a_{H,j}, \tag{9}$$

$$Fg(\downarrow) = \frac{1}{H-1}\sum_{j=1}^{H-1} \max_{\tau \in 1,\ldots H-1} a_{\tau,j} - a_{H,j}. \tag{10}$$

Implementation Details For fair comparisons, our BDT-SMT builds on the framework of CLS-ER [2] without modification to the backbone (i.e., a fully-connected network with two hidden layers for S-MNIST, and ResNet18 [12] without pretraining for the datasets S-CIFAR-10, S-CFIAR100 and S-Tiny-ImageNet), and adopts its reported experimental settings including batch size, minibatch size, training epochs, backward transport parameter for the four benchmarks. For the S-Mini-ImageNet, we use the EfficientNet-b2 [36] without pretraining as the backbone for all methods. The memory buffer size is set to 200, and the reservoir sampling [38] is adopted as the updating strategy for memory buffer to ensure sampling with the same probability. During training, Stochastic Gradient Descent (SGD) optimizer is adopted with the learning rate $\eta = 0.1$ for S-MNIST/S-CIFAR-10 and $\eta = 0.05$ for S-CIFAR-100/S-Tiny-ImageNet/S-Mini-ImageNet. For the forward transport momentum parameter $\hat{m}$, we set $\hat{m} = 0.99$ for S-MNIST and $\hat{m} = 0.999$ for S-CIFAR-10/S-CIFAR-100/S-Tiny-ImageNet/S-Mini-ImageNet. For the threshold k used in SMT, we set it to $\frac{1}{3}$ for all benchmarks. Code is available at https://github.com/S2VTouser/BDT-SMT.

4.2 Main Results

Table 1 shows the comparative results w.r.t. the state-of-the-arts in terms of average accuracy on the four benchmark datasets. Following [2], five baselines are used as competitors: ER [28], GEM [21], iCaRL [27], DER++ [6] and CLS-ER [2]. Furthermore, three latest methods are included as additional competitors: ER-ACE [7], SCoMMER [33] and ESMER [32]. Regarding the five baselines, except S-CIFAR-100 on which we re-implement all of them, the results on other datasets are directly reported from [2]. As for the three latest methods, we re-implement them on all datasets with their released code. Note that JOINT is the upper bound which indicates that the data from all tasks are used for training together instead of sequential training, while SGD is the lower bound which indicates that all tasks are sequentially trained with fine-tuning.

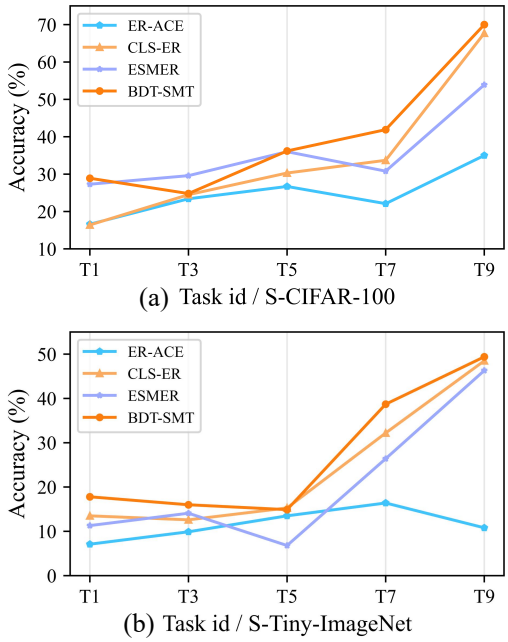

Figure 4: Detailed accuracy comparison results obtained by the final model on each previous task over the (a) S-CIFAR-100 and (b) S-Tiny-ImageNet datasets.

From Table 1, we can observe that: **(1)** Compared with the state-of-the-arts, our BDT-SMT shows superior performance by achieving the highest average accuracy across all datasets, indicating the effectiveness of our BDT-SMT for Class-IL. **(2)** Our BDT-SMT outperforms the second best ESMER by an average value 1.42% on all five datasets, and outperforms the third best CLS-ER by an average value 2.07% on all datasets. Particularly, our BDT-SMT surpasses ESMER by 2.33% on S-CIFAR-100 and 1.94% on S-Tiny-ImageNet.

Table 2: Comparative results obtained by changing the buffer size ($|M| = 200 \rightarrow 100, 50$) or the length of task sequences (Tasks Num) ($H = 10 \rightarrow 20$) on S-CIFAR-100 and S-Tiny-ImageNet. The average accuracy is reported over ten independent runs.

Method	Buffer Size	Tasks Num	S-CIFAR-100		S-Tiny-ImageNet	
Buffer Size	–	–	$\|M\|$=50	$\|M\|$=100	$\|M\|$=50	$\|M\|$=100
ER [28]	varied	H=10	10.23 (±0.30)	11.80 (±0.18)	8.11 (±0.08)	8.18 (±0.08)
DER++ [6]	varied	H=10	13.16 (±0.32)	14.80 (±1.71)	8.75 (±1.09)	10.42 (±0.44)
CLS-ER [2]	varied	H=10	22.80 (±0.48)	27.91 (±0.65)	14.34 (±0.58)	17.53 (±0.88)
SCoMMER [33]	varied	H=10	15.48 (±0.40)	23.61 (±1.20)	5.28 (±0.71)	10.33 (±0.55)
ESMER [32]	varied	H=10	21.70 (±1.00)	28.22 (±0.89)	13.74 (±0.90)	18.12 (±0.23)
BDT-SMT (ours)	varied	H=10	**24.45 (±0.63)**	**31.08 (±0.63)**	**15.63 (±0.51)**	**19.33 (±0.70)**
Tasks Num	–	–	H=10	H=20	H=10	H=20
ER [26]	$\|M\|$=200	varied	14.78 (±0.67)	14.61 (±0.49)	8.49 (±0.16)	4.82 (±0.19)
DER++ [6]	$\|M\|$=200	varied	26.40 (±1.17)	19.30 (±1.08)	10.96 (±1.17)	8.75 (±0.77)
CLS-ER [2]	$\|M\|$=200	varied	35.39 (±1.15)	22.19 (±1.90)	23.47 (±0.80)	15.99 (±0.88)
SCoMMER [33]	$\|M\|$=200	varied	31.75 (±1.39)	23.52 (±0.48)	16.61 (±0.46)	11.21 (±0.05)
ESMER [32]	$\|M\|$=200	varied	35.72 (±0.25)	27.25 (±0.52)	23.37 (±0.11)	10.86 (±0.69)
BDT-SMT (ours)	$\|M\|$=200	varied	**38.05 (±0.25)**	**28.11 (±0.38)**	**25.31 (±0.29)**	**18.00 (±0.52)**

Table 3: Ablation study results for our BDT-SMT.

Method	S-CIFAR-10		S-CIFAR-100	
	Acc (↑)	Fg (↓)	Acc (↑)	Fg (↓)
Base (CLS-ER)	66.19 (±0.75)	29.01 (±3.25)	35.39 (±1.15)	35.58 (±1.35)
Base+BDT	67.82 (±1.78)	20.47 (±5.49)	36.89 (±0.67)	**30.84 (±1.52)**
Base+BDT+SMT	**70.19 (±1.13)**	**19.54 (±3.60)**	**38.05 (±0.25)**	33.08 (±0.52)

Table 4: Comparative results of computational cost between the CLS-ER [2] and our BDT-SMT.

Method	Compute Time (s)		Storage (MiB)	
	S-CIFAR-10	S-CIFAR-100	S-CIFAR-10	S-CIFAR-100
CLS-ER	1571	807	2667	2669
BDT-SMT	1656	806	2745	2747

Additionally, BDT-SMT surpasses CLS-ER by 4.0% on S-CIFAR-10 and 2.61% on S-CIFAR-100. The obtained results provide a direct evidence that our BDT-SMT effectively acquires more new knowledge while consolidating the old knowledge, which yields significant benefits in enhancing model performance. Moreover, Figure 4 shows detailed comparison results obtained by the final model on each previous task (take S-CIFAR-100 and S-Tiny-ImageNet as examples). We can see more intuitively that compared to other methods, our BDT-SMT can make a better balance between the learned old knowledge (e.g., $T_1(T1), T_3, T_5$) and new knowledge (T_7, T_9), i.e., retaining as much old knowledge as possible while reducing the damage to the learning of new knowledge, thereby improving the overall performance of the model (corresponding to Figure 1). This is the fundamental goal of continual learning.

To further show the outstanding ability of our BDT-SMT, we conduct experiments under more strict and challenging scenarios: smaller memory buffer sizes ($|M|$, $|M| = 200 \rightarrow 100, 50$) and longer task sequences (H, $H = 10 \rightarrow 20$). The comparative results on S-CIFAR-100 and S-Tiny-ImageNet are shown in Table 2. The five comparative methods include ER [28], DER++ [6], CLS-ER [2], SCoMMER [33] and ESMER [32]. We implement these methods and ours by adopting the same experimental hyperparameters as in Table 1. It can be observed that: our BDT-SMT consistently achieves optimal results on both datasets under the two scenarios, indicating the superior generalization ability of our BDT-SMT in Class-IL. This ability allows the model to maintain excellent performance even with smaller buffer sizes and longer task sequences.

4.3 Ablation Study

To show the impact of proposed novel components (i.e., the BDT strategy as well as the SMT mechanism) on the performance of our BDT-SMT, we conduct ablative experiments on S-CIFAR-10 and S-CIFAR-100. We first take the originally followed method CLS-ER [2] as the baseline (denoted as Base). Then, on the basis of Base, we add the BDT strategy (denoted as Base+BDT). Last, we add the SMT mechanism (denoted as Base+BDT+SMT), i.e., our full BDT-SMT.

The ablative results are shown in Table 3. It can be clearly seen that: **(1)** When the BDT strategy is applied, the average accuracy is improved over Base, and especially the forgetting is greatly reduced (8.54% on S-CIFAR-10 and 4.74% on S-CIFAR-100). These gains strongly prove the effectiveness of the BDT strategy in retaining old knowledge and mitigating the forgetting. **(2)** When the SMT mechanism is also applied, a further improvement in average accuracy is observed (2.37% on S-CIFAR-10 and 1.16% on S-CIFAR-100) in comparison to Base+BDT. The improvement shows the effectiveness of SMT in better acquiring new knowledge, which contributes to the overall enhancement of model performance. Meanwhile, we can see that the average forgetting on S-CIFAR-100 increases, which may be due to lower forgetting caused by the lower maximum accuracy obtained on the previous tasks when only BDT is applied (see Eq. (9)). It is important to notice that the average accuracy is the primary metric to measure the continual learning performance. Furthermore, compared with Base, the average accuracy and average forgetting are significantly improved and decreased respectively (improved by 4.0% for Acc and decreased by 9.47% for Fg on S-CIFAR-10, improved by 2.66% for Acc and decreased by 2.5% for Fg on S-CIFAR-100), demonstrating that the proposed BDT and SMT have significant contributions to the improvement of model performance. Moreover, these results show the complementarity of the two proposed components, which is highly instructive for developing more advancing continual learning methods. Additionally, in Table 4 we show the computational costs between our BDT-SMT and Base (CLS-ER) in terms of the average computation time per task (Compute Time) and the total memory storage requirements (Storage). The similar computational costs between these two methods further underscore the efficiency of our method.

Considering the core role of both BDT and SMT, we conduct experiments to analyze the effect of two crucial hyperparameters,

Table 5: Comparative results of SMT-Mean and Mean-ER using a single semantic memory model under the Class-IL setting.

Method	Buffer Size	S-MNIST	S-CIFAR-10	S-CIFAR-100	S-Tiny-ImageNet	S-Mini-ImageNet
JOINT (upper bound)	–	95.57 (±0.24)	92.20 (±0.15)	70.55 (±0.91)	59.99 (±0.19)	51.40 (±0.31)
SGD (lower bound)	–	19.60 (±0.04)	19.62 (±0.05)	9.32 (±0.06)	7.92 (±0.26)	8.08 (±0.06)
Mean-ER	200	88.32 (±0.65)	61.88 (±2.43)	29.45 (±1.17)	17.68 (±1.65)	13.79 (±0.78)
SMT-Mean	200	**89.33 (±0.36)**	**64.06 (±1.32)**	**31.22 (±0.88)**	**23.68 (±0.39)**	**15.81 (±0.45)**
BDT-SMT	200	**89.99 (±0.27)**	**70.19 (±1.13)**	**38.05 (±0.25)**	**25.31 (±0.29)**	**20.82 (±0.60)**

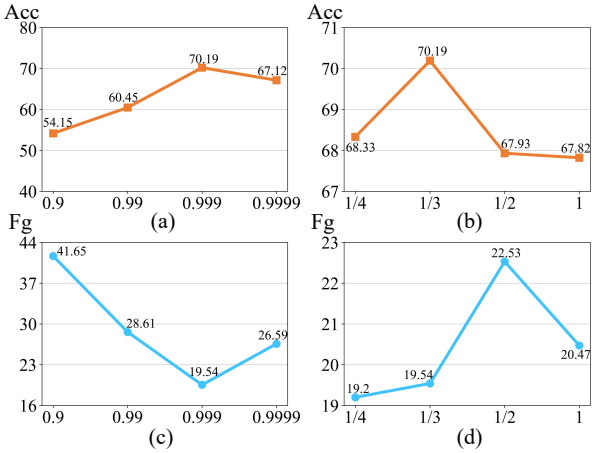

Figure 5: Comparative results of our BDT-SMT with different hyperparameters $\hat{m}$ ((a,c), with $k = \frac{1}{3}$ fixed) and k ((b,d), with $\hat{m} = 0.999$ fixed) on S-CIFAR-10.

namely $\hat{m}$ and k, on the performance of our BDT-SMT. We keep other hyperparameters fixed, to explore the forward transport momentum parameter $\hat{m} \in \{0.9, 0.99, 0.999, 0.9999\}$ and the threshold parameter $k \in \{1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}\}$. Figure 5 shows the comparative results for average accuracy (Acc) and average forgetting (Fg) on S-CIFAR-10 across ten independent runs with different $\hat{m}$ and k, respectively. It can be clearly seen that: **(1)** When $\hat{m}$ is 0.999, our BDT-SMT obtains the highest accuracy and the lowest forgetting. When $\hat{m}$ is too small or too large, the accuracy and forgetting tend to be compromised since transferring too much information of old parameters would affect the learning of new knowledge, and transferring too little would affect the consolidation of old knowledge. **(2)** When k is $\frac{1}{3}$, our BDT-SMT achieves the best overall performance even if the forgetting is slightly higher compared with $k = \frac{1}{4}$. The results of hyperparameter experiments on other datasets are similar to this, thus we finally set the forward transport momentum parameter $\hat{m} = 0.999$ on almost all benchmarks, and set the threshold parameter $k = \frac{1}{3}$ on all benchmarks.

4.4 Further Evaluation

To further validate the effectiveness of our proposed components (i.e., the BDT strategy and the SMT mechanism) for improving model performance in Class-IL, we conduct extra experiments by employing a single semantic memory model (denoted as SMT-Mean), and compare its performance against the baseline Mean-ER described in [2], which also utilizes a single semantic memory model. Table 5 shows the comparative results between SMT-Mean and Mean-ER on all five benchmarks. It can be observed that: our SMT-Mean significantly outperforms Mean-ER in all cases. More importantly, SMT-Mean can match or even exceed the performance of some classic methods in Table 1 (e.g., iCaRL [27] and DER++ [6]).

Table 6: Comparative results with different PIE algorithms.

Method	Buffer Size	S-CIFAR-10	S-CIFAR-100
PIE-EWC	200	68.01 (±0.68)	36.77 (±0.10)
PIE-SI (ours)	200	**70.19 (±1.13)**	**38.05 (±0.25)**

Particularly, the accuracy of our SMT-Mean slightly exceeds that of CLS-ER/ESMER [32] on S-Tiny-ImageNet. These results fully demonstrate the effectiveness of proposed two components, which are extremely beneficial/instructive for continual learning. Furthermore, our BDT-SMT exhibits superior performance over SMT-Mean on all benchmarks, indicating that the complementary application of two semantic memory models is more beneficial to enhancing the model performance compared to using only a single one.

To verify that the Parameter Importance Evaluation (PIE) algorithm designed according to SI [40] is more applicable to our BDT-SMT, we introduce the PIE algorithm designed according to EWC [16] into our method for comparative analysis. Here, we denote these methods as PIE-EWC and PIE-SI (ours) respectively. For fair comparisons, the PIE-EWC adopts the same experimental parameter settings as ours. Table 6 shows the average accuracy comparison results over ten independent runs. It can be observed that the PIE-SI achieves the highest accuracies on both datasets, significantly outperforming the PIE-EWC. This suggests the superior applicability of the PIE algorithm designed according to SI in our work. Moreover, we can see that the results obtained using the PIE-EWC on both datastes are comparable or even better than those obtained using the state-of-the-arts in Table 1, which further illustrates the effectiveness of our proposed components.

5 CONCLUSION

In this paper, we have proposed a novel CLS-based method termed BDT-SMT to acquire more new knowledge from new tasks while consolidating old knowledge, thereby significantly enhancing Class-IL for image classification. To effectively mitigate the forgetting, we first devise a bidirectional transport strategy between old and new models, which is quite different from the latest works [2, 32, 33] with only one unidirectional process (i.e., backward transport). Moreover, to ensure that the representation of new knowledge is not impaired by the old knowledge during forward transport, we design a selective momentum mechanism to selectively update parameters with the evaluation of parameters importance. Extensive experiments on five benchmark datasets demonstrate that our BDT-SMT significantly outperforms the state-of-the-arts under the Class-IL setting. Since the proposed BDT and SMT have a high flexibility/generalizability, we will explore how to apply them to other continual learning settings and even (momentum-based) contrastive learning settings in our ongoing research.

ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China (62376274). Zhiwu Lu is the corresponding author.

REFERENCES

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. 2018. Memory aware synapses: Learning what (not) to forget. In *European Conference on Computer Vision (ECCV)*. 139–154.

[2] Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. 2022. Learning fast, learning slow: A general continual learning method based on complementary learning system. *arXiv preprint arXiv:2201.12604* (2022).

[3] Arijit Banerjee and Vignesh Iyer. 2015. Cs231n Project report-tiny imagenet challenge.

[4] Eden Belouadah and Adrian Popescu. 2019. Il2m: Class incremental learning with dual memory. In *IEEE/CVF International Conference on Computer Vision (ICCV)*. 583–592.

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

[6] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. 2020. Dark experience for general continual learning: a strong, simple baseline. *Advances in Neural Information Processing Systems (NeurIPS)* 33 (2020), 15920–15930.

[7] Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. 2022. New insights on reducing abrupt representation change in online continual learning. *arXiv preprint arXiv:2203.03798* (2022).

[8] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2018. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420* (2018).

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 248–255.

[10] Enrico Fini, Victor G Turrisi Da Costa, Xavier Alameda-Pineda, Elisa Ricci, Karteek Alahari, and Julien Mairal. 2022. Self-supervised models are continual learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 9621–9630.

[11] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems (NeurIPS)* 33 (2020), 21271–21284.

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

[13] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

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

[15] Minsoo Kang, Jaeyoo Park, and Bohyung Han. 2022. Class-Incremental Learning by Knowledge Distillation with Adaptive Feature Consolidation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 16071–16080.

[16] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *National Academy of Sciences* 114, 13 (2017), 3521–3526.

[17] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases* 1, 4 (2009).

[18] Dharshan Kumaran, Demis Hassabis, and James L McClelland. 2016. What learning systems do intelligent agents need? Complementary learning systems theory updated. *Trends in Cognitive Sciences* 20, 7 (2016), 512–534.

[19] Yann LeCun, Corinna Cortes, and CJ Burges. 2010. MNIST handwritten digit database. AT&T Labs [Online]. *yann. lecun. com/exdb/mnist* (2010).

[20] Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 40, 12 (2017), 2935–2947.

[21] David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. *Advances in Neural Information Processing Systems (NeurIPS)* 30 (2017).

[22] Arun Mallya and Svetlana Lazebnik. 2018. Packnet: Adding multiple tasks to a single network by iterative pruning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 7765–7773.

[23] Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*. Vol. 24. Elsevier, 109–165.

[24] Quang Pham, Chenghao Liu, and Steven Hoi. 2021. Dualnet: Continual learning, fast and slow. *Advances in Neural Information Processing Systems (NeurIPS)* 34 (2021), 16131–16144.

[25] Quang Pham, Chenghao Liu, and Steven CH Hoi. 2022. Continual Learning: Fast and Slow. *arXiv preprint arXiv:2209.02370* (2022).

[26] Roger Ratcliff. 1990. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review* 97, 2 (1990), 285.

[27] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2001–2010.

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

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* 115 (2015), 211–252.

[30] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive neural networks. *arXiv preprint arXiv:1606.04671* (2016).

[31] Fahad Sarfraz, Elahe Arani, and Bahram Zonooz. 2021. Knowledge distillation beyond model compression. In *International Conference on Pattern Recognition (ICPR)*. 6136–6143.

[32] Fahad Sarfraz, Elahe Arani, and Bahram Zonooz. 2023. Error Sensitivity Modulation based Experience Replay: Mitigating Abrupt Representation Drift in Continual Learning. *arXiv preprint arXiv:2302.11344* (2023).

[33] Fahad Sarfraz, Elahe Arani, and Bahram Zonooz. 2023. Sparse coding in a dual memory system for lifelong learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 37. 9714–9722.

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

[35] Dhairyya Singh, Kenneth A Norman, and Anna C Schapiro. 2022. A model of autonomous interactions between hippocampus and neocortex driving sleep-dependent memory consolidation. *National Academy of Sciences* 119, 44 (2022), e2123432119.

[36] Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML)*. PMLR, 6105–6114.

[37] Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in Neural Information Processing Systems (NeurIPS)* 30 (2017).

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

[39] Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. 2022. Foster: Feature boosting and compression for class-incremental learning. In *European Conference on Computer Vision (ECCV)*. Springer, 398–414.

[40] Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. In *International Conference on Machine Learning (ICML)*. PMLR, 3987–3995.

[41] Da-Wei Zhou, Qi-Wei Wang, Han-Jia Ye, and De-Chuan Zhan. 2022. A model or 603 exemplars: Towards memory-efficient class-incremental learning. *arXiv preprint arXiv:2205.13218* (2022).

[42] Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. 2021. Co-transport for class-incremental learning. In *the 29th ACM International Conference on Multimedia*. 1645–1654.