Decoupling Learning and Remembering: a Bilevel Memory Framework with Knowledge Projection for Task-Incremental Learning

Wenju Sun[1] Qingyong Li[1,2,3] Jing Zhang[1] Wen Wang[1] Yangli-ao Geng[1,2,3,*]
[1]Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University
[2]Frontiers Science Center for Smart High-speed Railway System, Beijing Jiaotong University
[3]The Key Laboratory of Cognitive Computing and Intelligent Information Processing of Fujian Education Institutions, Wuyi University
{SunWenJu,liqy,j_zhang,wangwen,gengyla}@bjtu.edu.cn

Abstract

*The dilemma between plasticity and stability arises as a common challenge for incremental learning. In contrast, the human memory system is able to remedy this dilemma owing to its multi-level memory structure, which motivates us to propose a **B**ilevel **M**emory system with **K**nowledge **P**rojection (BMKP) for incremental learning. BMKP decouples the functions of learning and remembering via a bilevel-memory design: a working memory responsible for adaptively model learning, to ensure plasticity; a long-term memory in charge of enduringly storing the knowledge incorporated within the learned model, to guarantee stability. However, an emerging issue is how to extract the learned knowledge from the working memory and assimilate it into the long-term memory. To approach this issue, we reveal that the parameters learned by the working memory are actually residing in a redundant high-dimensional space, and the knowledge incorporated in the model can have a quite compact representation under a group of pattern basis shared by all incremental learning tasks. Therefore, we propose a knowledge projection process to adaptively maintain the shared basis, with which the loosely organized model knowledge of working memory is projected into the compact representation to be remembered in the long-term memory. We evaluate BMKP on CIFAR-10, CIFAR-100, and Tiny-ImageNet. The experimental results show that BMKP achieves state-of-the-art performance with lower memory usage*[1].

1. Introduction

In an ever-changing environment, intelligent systems are expected to learn new knowledge incrementally without forgetting, which is referred to as incremental learning (IL) [8, 18]. Based on different design principles, many incremental learning methods have been proposed [16, 19, 24]. Among them, memory-based models are leading the way in performance and have attracted enormous attention [3, 17, 26, 33].

The core idea of memory-based methods is to utilize partial old-task information to guide the model to learn without forgetting [20]. As illustrated in Figure 1 (a), memory-based methods typically maintain a memory to store the old-task information. According to the type of stored information, memory-based methods further fall into two categories: rehearsal-based and gradient-memory-based. Rehearsal-based methods [3, 17, 28, 29, 39] keep an exemplar memory (or generative model) to save (or generate) old-task samples or features, and replay them to recall old-task knowledge when learning new tasks. However, the parameter fitting of new tasks has the potential to overwrite the old-task knowledge, especially when the stored samples are unable to accurately simulate old-task data distributions, leading to low stability. Gradient-memory-based methods [26, 33] maintain a memory to store the gradient directions that may interfere with the performance of old tasks, and only update the learning model with the gradients that are orthogonal to the stored ones. Although the gradient directions restriction guarantees stability, this restriction may prevent the model from being optimized toward the right direction for a new task, which would result in low plasticity. Therefore, both of these two types of methods suffer the low plasticity or stability. The reason is that their model is in charge of both learning new task knowledge and maintaining old task knowledge. The limited model capacity will inevitably lead to a plasticity-stability trade-off in the face of a steady stream of knowledge, i.e., plasticity-stability dilemma [21].

In contrast, human brains are known for both high plasticity and stability for incremental learning, owing to its multi-level memory system [7]. Figure 1 (b) illustrates how a human brain works in the classic Aktinson-Shiffrin human

*Corresponding author: Yangli-ao Geng (gengyla@bjtu.edu.cn).
[1]The code is available at https://github.com/SunWenJu123/BMKP

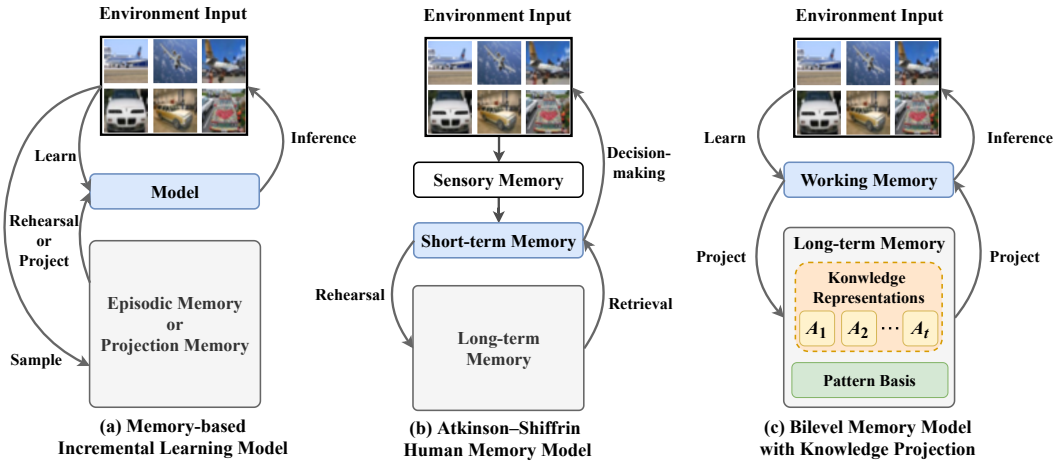

Figure 1. Diagram of memory-based incremental learning method (a), Atkinson-Shiffrin human memory model [30] (b), and the architecture of the proposed BMKP (c).

memory model [30]. Inspired by the mechanism of human memory, this paper proposes a **B**ilevel **M**emory model with **K**nowledge **P**rojection (BMKP) for incremental learning. As illustrated in Figure 1 (c), BMKP adopts a bilevel-memory design, including a working memory (corresponds to the short-term memory of the human brain) and a long-term memory. The working memory is implemented as a neural network responsible for adaptively learning new knowledge and inference. The long-term memory is in charge of steadily storing all the learned knowledge. Similar to the human memory, this bilevel-memory structure endows BMKP with both high plasticity and stability by decoupling the functions of learning and remembering.

An emerging issue for this bilevel memory framework is how to extract the learned knowledge from the working memory and assimilate it into the long-term memory. In the working memory, the knowledge is represented as the trained parameters in a high-dimensional space, which we call **Parameter Knowledge Space (PKS)**. However, this space is usually overparameterized [6], implying that the knowledge representation in PKS is loosely organized. Therefore, instead of directly storing the learned parameters, we propose to recognize the underlying common patterns, and further utilize these patterns as the basis to represent the parameters. Specifically, we define the space spanned by these pattern basis as the **Core Knowledge Space (CKS)**, in which the knowledge can be organized in a quite compact form without loss of performance. Based on these two knowledge spaces, we propose a knowledge projection process to adaptively maintain a group of CKS pattern basis shared by all incremental learning tasks, with which the loosely organized model knowledge in PKS can be projected into CKS to obtain the compact knowledge representation. The compact representation, instead of the raw model knowledge, is transferred to the long-term memory for storing.

The contributions of this work are summarized as follows:

- Inspired by the multi-level human memory system, we propose a bilevel-memory framework for incremental learning, which benefits from both high plasticity and stability.
- We propose a knowledge projection process to project knowledge from PKS into compact representation in CKS, which not only improves memory utilization efficiency but also enables forward knowledge transfer for incremental learning.
- A representation compaction regularizer (Eq. (4)) is designed to encourage the working memory to reuse previously learned knowledge, which enhances both the memory efficiency and the performance of BMKP.
- We evaluate BMKP on CIFAR-10, CIFAR-100, and Tiny-ImageNet. The experimental results show that BMKP outperforms most of state-of-the-art baselines with lower memory usage.

2. Related Work

Incremental learning requires models to learn new knowledge incrementally without forgetting [18]. Specifically, this work focuses on task incremental learning, which is known as a conventional setting in IL research [32]. The task label of a query sample is available in many scenarios, such as cross-camera recognition and multi-lingual translation tasks. Moreover, the Task-IL setting allows us to simplify the problem scenario and devote more effort to the targeted challenge, the plasticity-stability dilemma.

According to the mechanism for preventing forgetting, incremental learning methods can be categorized into three

classes: regularization-based methods, expansion-based methods, and memory-based methods. Regularization-based methods alleviate forgetting with regularization terms. Some methods apply knowledge distillation to regularize the activations of neural networks [10, 16, 39]. Other methods measure the importance of network parameters and then limit their changes when learning new tasks [1, 14, 38]. Expansion-based methods dynamically expand the network capacity during incremental learning. Progressive neural networks (PGN) [25] creates a specific network for each task and transfers knowledge among different networks through horizontal connections. However, the memory usage will increase linearly with the number of learned tasks. To mitigate the memory usage, Dynamically Expandable Networks (DEN) [37], Reinforced Continual Learning (RCL) [34], and Additive Parameter Decomposition (APD) [36] only expand the width of networks when capacity is insufficient. Memory-based methods keep an extra memory to store old-task sample information, including rehearsal-based methods and gradient-memory-based methods. Rehearsal-based methods [3, 4, 29, 35] keep an exemplar memory or generative model and replay old-task (pseudo) samples or features when learning new tasks to prevent forgetting. Gradient-memory-based methods [5, 17, 26, 33] calculate or keep the gradient directions that can interfere with the model performance for old tasks, and constrain the gradient descent direction to be orthogonal to them. Like gradient-memory-based methods, BMKP also stores the model information for the old tasks, but BMKP chooses to store the projected knowledge instead of the gradient information in light of the human memory mechanism.

3. Preliminary

3.1. Human Memory Mechanism

Human beings are skilled in learning new knowledge incrementally owing to their delicate brain structure and effective memory mechanism [7, 30]. As shown in Figure 1 (b), the classic Atkinson-Shiffrin human memory model [30] deems that human memory consists of three separate components: sensory memory, short-term memory, and long-term memory. The sensory memory is responsible for caching signals caused by environmental stimuli. Compared with the sensory memory, the short-term memory has a larger capacity and a longer storage duration, which allows it to process the cached signals into information to facilitate human decision-making and behavior. After that, the brain performs a rehearsal step to re-organize important information into compact knowledge, which is then transferred to the long-term memory for storing. The long-term memory has the largest capacity and the longest storage duration among the three components, where the stored knowledge can be retrieved back to the working memory as the brain need. This multi-level memory structure endows the human brain with three advantages: (i) **plasticity** to learn new knowledge, (ii) **stability** to maintain old knowledge, (iii) **efficient memory utilization** to store tremendous knowledge with compact representations. These advantages ensure the incremental learning ability of human beings.

3.2. Problem Definition

This work focuses on the task incremental learning setting (Task-IL) [32], where a model is required to learn knowledge from a stream of datasets of T tasks: D_1, D_2, ..., D_T. Specifically, during training the task t, only the dataset $D_t = (X_t, Y_t)$ is available, where X_t and Y_t denote the feature set and the label set, respectively. The knowledge for dealing with different tasks is generally distinctive. In the supervised classification context, the classes to be recognized in different tasks are disjoint, i.e., $Y_{t_1} \cap Y_{t_2} = \emptyset$ for $t_1 \neq t_2$. During testing, the model is evaluated on all the learned tasks, where the task identification, along with each test sample, is provided to the model.

4. Bilevel Memory Framework

4.1. Overview

Motivated by the human memory mechanism, we propose a bilevel memory framework for incremental learning, including the following two memory units.

Working memory is responsible for adaptively learning new knowledge and inference. This component is implemented as a L-layer neural network, it receives an input sample x and outputs a prediction $\hat{y}$ for x:

$$\hat{y} = f(x; W), \tag{1}$$

where $W = \{W^1, ..., W^L\}$ denotes the network parameters of all the L layers. To be more specific, for the l-th layer with parameter $W^l \in \mathbb{R}^{p_{l+1} \times p_l}$, where p_l denotes the input dimension of the l-th layer, features are extracted based on the output from the last layer:

$$Z^l = W^l X^l \quad X^{l+1} = \sigma(Z^l), \tag{2}$$

where $X^l \in \mathbb{R}^{p_l \times n}$ is the feature extracted by l-th layer, n denotes the number of training samples, and σ is a parameter-free non-linear unit such as the ReLU activation. Noting that any convolution layer can be efficiently transformed into a fully connected layer[2], we assume W^l to be the parameter matrix of a fully connected layer in this paper without loss of generality.

Long-term memory is in charge of storing all learned knowledge. Also, the knowledge can be reloaded into the working memory for inference. A trivial idea is storing parameters learned by working memory directly, known as

[2] See Figure 1 in Appendix for an illustration.

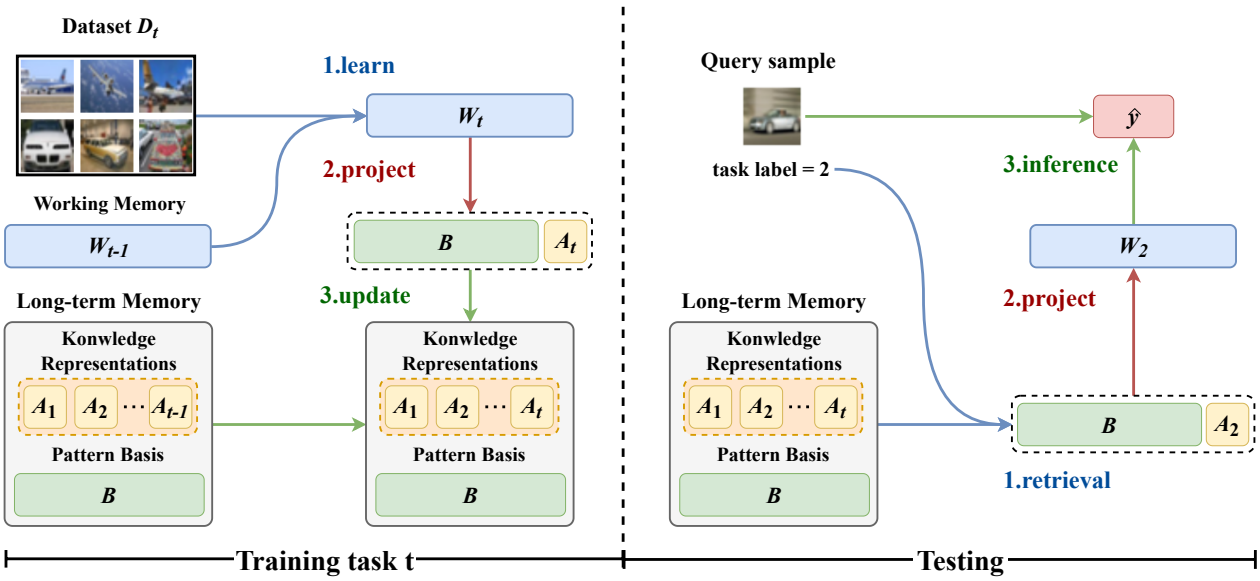

Figure 2. Diagram of training (left) and testing (right) processes of BMKP.

single-task learning [26]. However, as the number of tasks increases, the storage overhead of this simple idea becomes prohibitive. Our solution is seeking a core low-dimensional subspace of the original model parameter space, such that the projection of knowledge (i.e., model parameters) into that subspace incorporates valuable information as much as possible but with much more compact representations. We elaborate on our idea in the following.

4.2. Knowledge Spaces

Given a neural network model, the knowledge contained in the model can be thought of as the ability to transfer a given input to the expected output, which can be further explicitly represented by the network parameters W in PKS. However, this trivial representation is quite loose since neural networks are usually overparameterized [6], implying that a lot of redundant dimensions may exist in W. Besides, as a common assumption for incremental learning [13], the knowledge from different tasks can be expected to share common underlying patterns. The above facts motivate us to identify more compact knowledge representations for the bilevel memory framework.

Observing Eq. (2), Z^l are generated by the linear combination of the columns of W^l. We can infer that only the part of W^l which falls into the column space of Z^l contributes the knowledge for extracting Z^l. To be more specific, let B^l be a group of basis of the space spanned by Z^l, the first equation in Eq. (2) leads to

$$\begin{aligned} Z^l = B^l {B^l}^\top Z^l &= B^l {B^l}^\top W^l X^l \\ &= \hat{W}^l X^l \\ &= B^l A^l X^l, \end{aligned} \tag{3}$$

where $\hat{W}^l = B^l {B^l}^\top W^l$ denotes the projection of W^l into the column space of Z^l and $A^l = {B^l}^\top W^l$ is the coefficient matrix of $\hat{W}^l$ under the basis of B^l.

Equation (3) indicates that $\hat{W}^l$ contains all the knowledge which processes given input X^l to the expected output Z^l, and the coefficient matrix A^l can be viewed as a compact knowledge representation of W^l. Therefore, we call the space $\hat{W}^l$ resides in (i.e., the space spanned by Z^l) as the **Core Knowledge Space (CKS)**, and the basis of CKS B^l as the pattern basis.

5. Bilevel Memory with Knowledge Projection

Based on the bilevel memory framework described in Section 4, a knowledge projection process is introduced as the bridge between the two memories, which together constitute the proposed BMKP. For ease of the ensuing presentation, let us recall some key notations introduced in Section 4.2. In the entire training process of BMKP, a group of pattern basis of CKS $B = \{B^1, ..., B^L\}$ is adaptively maintained and will be extended as needed, where B^l denoted the basis of the l-th network layer. With the basis B, the knowledge (i.e. parameters of a well-trained model) learned for the t-th task can be represented by $A_t = \{A_t^1, ..., A_t^L\}$.

As illustrated in the left part of Figure 2, the BMKP training process for a task t consists of three steps: work-

ing memory learning, knowledge projection, and long-term memory updating, which will be elaborated on in the following subsections.

5.1. Working Memory Learning

Following the principle of minimum energy consumption, brains always try to represent new knowledge by previously built patterns. Motivated by this principle, BMKP encourages the working memory to learn new knowledge with respect to the pattern basis B. Specifically, we expect that the learned parameters W can be represented by B as well as possible, which leads us to propose the following representation compaction regularizer:

$$L_{reg}(W) = \sum_{l=1}^{L} \frac{\text{Trace}\left((W^l - \tilde{W}^l)^\top (W^l - \tilde{W}^l)\right)}{\left\|(W^l - \tilde{W}^l)^\top (W^l - \tilde{W}^l)\right\|_1}, \quad (4)$$

where $\tilde{W}^l = B^l B^{l\top} W^l$ denotes the orthogonal projection of W^l into CKS. Intuitively, the numerator of Eq.(4) minimizes the Frobenius norm of the residual $W^l - \tilde{W}^l$, which actually encourages W^l to fall into CKS spanned by B_l so that the knowledge of old tasks can be transferred for the current task learning; On the other hand, the denominator maximizes the second moment of the residual, which essentially regularizes $W^l - \tilde{W}^l$ to be approximately low-rank (i.e., the rows of $W^l - \tilde{W}^l$ approximately lie in a low-dimensional subspace). We empirically find that this regularizer (With a proper weight) can not only reduce memory overhead but also enhance the performance of BMKP, as shown in Table 3.

Consequently, the working memory learning process for the task t can be formulated as

$$W_t \longleftarrow \arg\min_{W} L_{task}(W, D_t) + \lambda L_{reg}(W), \quad (5)$$

where L_{task} denotes the loss function for task t learning (e.g., cross-entropy loss for classification), and λ is a hyper-parameter weighting the regularizer.

5.2. Knowledge Projection

Next, BMKP pursues a compact representation in CKS for the learned knowledge W_t in PKS. However, directly projecting W_t into CKS may incur erratic knowledge loss and further downgrade the model performance. To overcome this drawback, we first properly extend the CKS basis B so that CKS can accommodate W_t well. Recall that in Eq.(3), we derive that the effective part of $\hat{W}_t^l$ falls into the column space of Z_t^l. This fact motivates us to build new pattern basis from the column space of Z_t^l. Specifically, we design the following basis updating process:

$$\begin{aligned} U_t^l \Sigma_t^l V_t^{l\top} &\longleftarrow \text{SVD}\left(Z_t^l - B^l B^{l\top} Z_t^l\right), \\ B^l &\longleftarrow \left[B^l \quad U_t^l\right], \end{aligned} \quad (6)$$

Algorithm 1: BMKP for Task Incremental Learning

Input: Datasets $\{D_1, \ldots, D_T\}$
Output: Knowledge representations $A = \{A_1, \ldots, A_T\}$; Pattern basis B

Initialize B;
$A \longleftarrow \emptyset$;
for $t = 1, \ldots, T$ **do**
  $W_t \longleftarrow$ Working_Memory_Learning(D_t, B)
  // Refer to Eq. (5)
  $A_t, B \longleftarrow$ Knowledge_Proj.(W_t, X_t, B)
  // Refer to Eq. (6) and (7)
  $A_t \longleftarrow$ Long-term_Memory_Updating(A_t, B)
  // Refer to Eq. (8)
  $A \longleftarrow A \cup \{A_t\}$;

where SVD$(\cdot)$ denotes the singular value decomposition operator. To keep the basis compact, we introduce the retained singular-value ratio[3] to make basis selection. Only the basis corresponding to the retained singular values is added. Notably, all the pattern basis in B^l are orthonormal due to the definition of singular vectors. Furthermore, the updating of B^l will not break this property since the new basis in U^l is guaranteed to be orthogonal to the existing one.

With the updated basis B^l, we can project W_t^l into CKS with little knowledge loss:

$$B^{l\top} W_t^l = A_t^l. \quad (7)$$

By applying knowledge projection to all the layers of W_t, we acquire the knowledge representation $A_t = \{A_t^1, \ldots, A_t^L\}$ and the updated pattern basis $B = \{B^1, \ldots, B^L\}$. It is clear that A_t provides a compact representation for W_t under the pattern basis B. Notably, the expansion of B may incur a dimension mismatching problem between B and the knowledge representations A_j for the previous task j $(j < t)$. Fortunately, as the expansion of B is strictly incremental, when we need to compose W_j from A_j and B, the dimension mismatching problem can be addressed by simply clipping the superfluous basis[4] (which are never used by A_j) from B.

5.3. Long-term Memory Updating

Through the knowledge projection, W_t is re-expressed by A_t with the pattern basis B. However, this re-expression may not be perfect since some minor basis are dropped through threshold selection, which may incur a performance degradation. We introduce a recall mechanism to handle this issue: re-training the task t with the A_t in CKS. This step

[3] It is defined as the ratio of the sum of retained singular values to the sum of total singular values.

[4] We provide more details in Section 3.2 in Appendix.

Table 1. Comparison results on several datasets. We report the average accuracy (%) over five runs with random seeds, and the higher is the better. (*) indicates the upper-bound model that is jointly trained with all tasks. (-) means that the result was unavailable, due to the intractable training time by our implementation or missing in the original paper. (†) implies its results are quoted from the original paper for those using the same dataset split with us. Remarkably, the difference in the result of CIFAR-100 between GPM and GPM† can be due to the different backbone choice: GPM adopts ResNet18 by our unified setting, while GPM† uses the 5-layer AlexNet in its original paper.

Methods	**Venue**	**CIFAR-10**	**CIFAR-100**	**Tiny-ImageNet**	**Average**
Joint*	-	98.07	91.18	82.01	90.42
LwF [16]	TPAMI2017	91.91±0.7	63.78±4.3	58.61±1.8	71.43
SI [38]	ICML2017	76.15±2.6	62.21±2.6	60.91±1.3	66.42
DGR [29]	NIPS2017	91.06±7.4	44.53±2.5	-	-
GEM [17]	NIPS2017	85.14±2.1	62.80±2.7	44.66±1.7	64.20
oEWC [27]	ICML2018	64.17±4.8	38.40±1.9	31.91±0.9	44.83
LwM [9]	CVPR2019	78.01±0.8	68.88±0.9	45.57±0.2	64.15
DI [35]	CVPR2020	94.46±0.6	68.43±2.1	66.12±0.9	76.34
DER [3]	NIPS2020	93.13±0.3	73.26±1.3	51.22±1.5	72.54
DER++ [3]	NIPS2020	93.71±0.4	74.86±1.1	53.00±0.4	73.86
DER++† [3]	NIPS2020	93.88±0.5	-	51.91±0.7	-
HAL [4]	AAAI2021	82.34±1.5	43.91±3.6	-	-
PASS [39]	CVPR2021	86.07±0.2	77.30±0.4	62.87±0.4	75.41
GPM [26]	ICLR2021	86.58±0.9	70.93±0.9	59.84±0.2	72.45
GPM† [26]	ICLR2021	-	72.48	-	-
Adam-NSCL [33]	CVPR2021	87.23±0.4	65.69±0.2	59.98±0.7	70.97
CLS-ER [2]	ICLR2022	93.53±0.3	72.11±0.5	57.36±0.7	74.33
WSN [12]	ICML2022	92.99±0.4	**81.10±0.7**	67.50±0.7	80.53
CF-IL† [23]	ICLR2022	93.12	-	67.42	-
FAS [22]	ICLR2022	90.89±1.3	70.89±0.6	60.10±0.2	73.96
BMKP (ours)	-	**94.49±0.2**	79.62±0.8	**70.36±0.2**	**81.49**

can be formulated as

$$A_t \longleftarrow \arg\min_{A_t} L_{task}(BA_t, D_t). \tag{8}$$

After retraining, we store A_t and B in long-term memory, and finish the learning process for the task t. Algorithm 1 summarizes the learning process of BMKP.

During testing, as illustrated in the right part of Figure 2, BMKP first retrieves the knowledge representation A_t according to the task label of the query sample, and re-compose W_t based on A_t and B. Then, BMKP loads W_t into working memory and conducts inference.

6. Experiments

6.1. Settings

Datasets: We evaluate BMKP on three image classification datasets, including **5-split CIFAR-10** [15], **10-split CIFAR-100** [15], **10-split Tiny-ImageNet** [31]. 5-split CIFAR-10 is constructed by splitting 10 classes of CIFAR-10 into 5 tasks with 2 classes per task. Similarly, 10-split CIFAR-100 is constructed by splitting 100 classes of CIFAR-100 into 10 tasks where each task has 10 classes, and 10-split Tiny-ImageNet is obtained by dividing Tiny-ImageNet into 10 tasks with 20 classes per task.

Baselines: We compare our method with various latest and classic incremental learning methods, including Learning without Forgetting (LwF) [16], Synaptic Intelligence (SI) [38], Deep Generative Replay (DGR) [29], Gradient Episodic Memory (GEM) [17], online Elastic Weight Consolidation (oEWC) [27], Learning without Memorizing (LwM) [9], DeepInversion (DI) [35], Dark Experience Replay (DER and DER++) [3], Prototype Augmentation and Self-Supervision (PASS) [39], Gradient Projection Memory (GPM) [26], Adam-NSCL [33], Hindsight Anchor Learning (HAL) [4], Complementary Learning System (CLS-ER) [2], Winning SubNetworks (WSN) [12], CF-IL [23], and Filter Atom Swapping (FAS) [22]. Besides, we also report the performance of a base model (Joint), which is trained by all task data jointly. Clearly, Joint does not follow the Task-IL setting, and its performance can be regarded as the upper bound of incremental learning methods. Notably, some baselines are designed for the Class-IL setting, for which the multi-head versions are applied for a fair comparison.

Performance metrics: Following [3, 17], we use the

Table 2. Ablation study of basis updating and retraining.

Methods	Split CIFAR-10	Split CIFAR-100	Split Tiny-ImageNet
BMKP w/o basis updating	79.44±2.7	43.00±1.9	28.27±1.1
BMKP w/o retraining	94.07±0.3	78.73±0.6	68.12±0.8
BMKP	94.49±0.2	79.62±0.8	70.36±0.2

classification accuracy (ACC) to evaluate the performance of all methods. To alleviate the influence of randomness in neural network training, we run all experiments five times with random seeds and report the average performance. Besides, we also report memory usage in megabyte (MB) of all baselines, including network parameters and extra storage (e.g., long-term memory for BMKP, exemplars for DER, and gradients for GPM and Adam-NSCL).

Implementation details: Considering the fairness of comparison, we compare the performance of all methods with similar memory utilization. As the size of extra memory for GPM, Adam-NSCL, and BMKP is related to the network width. Following [26], we apply a smaller version of ResNet18 [11] for these three methods. In contrast, all the other baselines employ the standard ResNet18 as the network backbone. Besides, rehearsal-based methods need additional memory to maintain exemplars. We provide them with a buffer that can keep up to 500 samples. The network parameters are optimized by the stochastic gradient descent (SGD) optimizer and iteratively updated 50, 100, and 100 epochs on CIFAR-10, CIFAR-100, and Tiny-ImageNet, respectively. Other hyperparameters are searched through a validation set obtained by sampling ten percent samples from the training set. For BMKP, the batch size is set to 128, and the learning rate is set to 0.05, 0.05, and 0.03 for three datasets, respectively. The balance weight λ is tuned from the range $[0, 1, 10, 100, 1000]$ and set to 10, 1, and 10 for CIFAR-10, CIFAR-100, and Tiny-ImageNet datasets, respectively. We tune the retained singular-value ratio (Cf. Eq. (6)) in $[95\%, 96\%, 97\%, 98\%, 99\%]$, and finally set it to 96%, 96% and 97% for the CIFAR10, CIFAR100, and Tiny-ImageNet, respectively. More details can be found in our supplementary code.

6.2. Comparison Result

We first evaluate BMKP on the 5-split CIFAR-10 task. As shown in the third column of Table 1, our method achieves the best performance with an average accuracy of 94.49%. Compared to the second-best DI, BMKP is 0.03% more accurate on CIFAR-10 and shows greater performance on the other two datasets, i.e., CIFAR-100 and Tiny-ImageNet. The results on the 10-split CIFAR-100 are reported in the fourth column of Table 1, which demonstrate that BMKP achieves the second-best performance and comparable accuracy with WSN. For the 10-split Tiny-ImageNet task, BMKP outperforms all comparison baselines and achieves 70.36% average accuracy, which is 2.86% higher than the second-best method WSN. In the last column of Table 1, we calculated the average performance on three datasets for all methods, and our BMKP shows the highest 81.49%, which is 0.96% higher than the second-best method. In conclusion, with the bilevel memory framework, BMKP ensures high plasticity and stability for incremental learning, and thus achieves state-of-the-art performance on all three benchmarks.

Table 3. Incremental learning results of BMKP with different λ (the regularizer weight, see Eq.(5)) on 10-split-Tiny-ImageNet.

λ	0	1	10	100	1000
ACC (%)	57.41	69.18	70.36	67.41	47.97
Memory (MB)	32.15	26.81	25.05	24.98	24.78

6.3. Ablation Study

We conduct an ablation study to investigate the effectiveness of basis updating and retraining. As shown in Table 2, both basis updating and retraining improve the performance of BMKP. Furthermore, the basis updating is more important than the retraining. It is reasonable since the basis updating extends the representation range of model parameters which highly affects the learning capacity for new tasks, while the retraining is a fine-tuning process that only refines the model performance.

6.4. Effectiveness of Representation Compaction Regularizer

To investigate the effectiveness of the representation compaction regularizer, we conduct experiments by setting different values of λ $(0, 1, 10, 100, 1000)$, and the results are shown in Table 3. As can be seen, when $\lambda = 0$, i.e., the representation compaction regularizer takes no effect, the model has the worst performance and the highest memory usage. As λ rises, the classification accuracy increases and reaches the peak (70.36%) when $\lambda = 10$. However, when λ becomes larger, the performance of the model gradually drops. This phenomenon reveals that the representation compaction regularizer can enhance the performance of BMKP by transferring more learned knowledge. Moreover, the memory usage of BMKP drops monotonously as λ increases, showing that the representation compaction regularizer can improve the

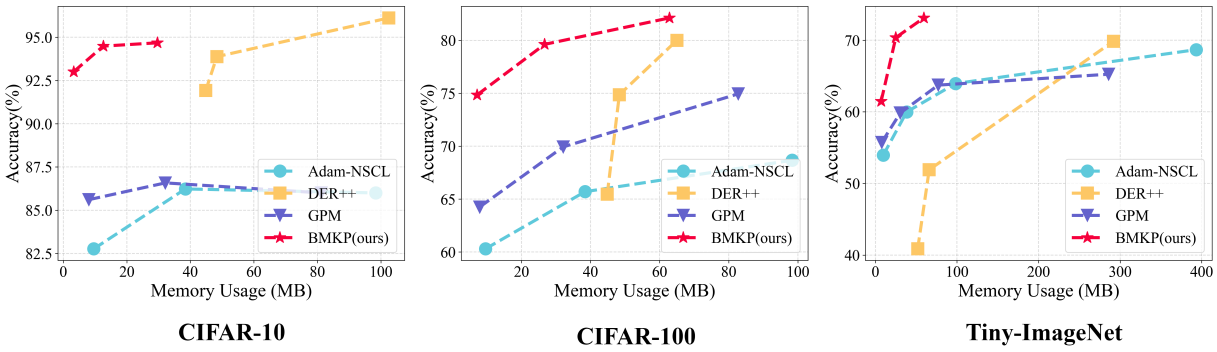

Figure 3. Diagram of ACC (%) over different memory usage (MB).

Table 4. Incremental learning results with different SVD ratio.

Dataset	Metrics	0.95	0.96	0.97	0.98
CIFAR-100	ACC(%)	78.48	79.62	80.14	81.50
	Mem(MB)	25.67	26.71	27.67	29.87
Tiny-ImageNet	ACC(%)	69.90	70.10	70.36	70.12
	Men(MB)	25.64	25.95	26.03	26.38

memory efficiency of BMKP since it encourages the model to represent new knowledge with previous-built patterns.[5]

6.5. Memory Efficiency Analysis

To study the memory efficiency of BMKP, we evaluate the performance of models with different memory usage. However, directly controlling the memory usage of different methods is impractical. As a comprise, we choose to adjust the number of stored samples for the rehearsal-based method (DER++) and scale the width of the network backbone for the other methods (GPM, Adam-NSCL, and BMKP). Figure 3 illustrates the results of methods on the three datasets. We notice that the performance of DER++ is greatly affected by memory size. Because the large memory size allows DER++ to keep more samples to prevent forgetting. However, when the available samples are limited, the performance of DER++ drops sharply as the few samples can not simulate the true distribution of data. Besides, as for BMKP, the larger memory ensures its better performance. More importantly, with the same memory usage, BMKP can perform the best accuracy on all three datasets. All of these experimental results prove that BMKP has high memory efficiency.

6.6. Influence of SVD ratio

We provide experimental results to show the effect of the SVD ratio (referred to as γ). As shown in Table 4, both accuracy and memory usage basically increase monotonically with γ, which is in line with our intuition that a larger γ retains more basis, and thus the model can be better reconstructed. We empirically choose $\gamma = 0.96$ for CIFAR100 and $\gamma = 0.97$ for Tiny-ImageNet to trade off accuracy and memory usage.

[5]We also found that the regularizer have different effects on various layers, and carefully tuning the weights may lead to better performance. See Section 5.8 in Appendix for a detailed analysis.

7. Conclusion

Inspired by the mechanism of human memory, we propose a bilevel memory model with knowledge projection (BMKP). BMKP applies a bilevel-memory framework, including a working memory responsible for adaptively learning new knowledge and inference, and a long-term memory charging for storing all learned knowledge. This structure decouples the functions of learning and maintaining knowledge, which guarantees high plasticity and stability. Moreover, motivated by the organization of human memory, BMKP introduces a knowledge projection step that serves as a bridge for the communication between these two memories, which re-organizes high-dimensional working memory parameters into compact task-specific knowledge representations and task-shared pattern basis. Nonetheless, there are some limitations of our model, such as missing backward knowledge transfer, relying on task labels, and growing memory usage with the number of learned tasks. We will address these limitations in our future work.

8. Acknowledgments

This work was supported in part by the Fundamental Research Funds for the Central Universities (Science and technology leading talent team project, Grant 2022JBQY007), the National Natural Science Foundation of China (Grant 62276019, U2034211, 62006017), the Open Project Program of The Key Laboratory of Cognitive Computing and Intelligent Information Processing of Fujian Education Institutions, Wuyi University, the China Postdoctoral Science Foundation (Grant 2022M721826).

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *European Conference on Computer Vision*, pages 139–154, 2018. 3

[2] Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. Learning fast, learning slow: a general continual learning method based on complementary learning system. In *International Conference on Learning Representations*, 2022. 6

[3] 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*, volume 33, pages 15920–15930, 2020. 1, 3, 6

[4] Arslan Chaudhry, Albert Gordo, Puneet Kumar Dokania, Philip Torr, and David Lopez-Paz. Using hindsight to anchor past knowledge in continual learning. In *AAAI Conference on Artificial Intelligence*, 2021. 3, 6

[5] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *International Conference on Learning Representations*, 2019. 3

[6] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. A survey of model compression and acceleration for deep neural networks. *arXiv preprint arXiv:1710.09282*, 2017. 2, 4

[7] Rosemary A Cowell, Morgan D Barense, and Patrick S Sadil. A roadmap for understanding memory: Decomposing cognitive processes into operations and representations. *Eneuro*, 6(4), 2019. 1, 3

[8] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1

[9] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2019. 6

[10] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *European Conference on Computer Vision*, pages 86–102, 2020. 3

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 7

[12] Haeyong Kang, Rusty John Lloyd Mina, Sultan Rizky Hikmawan Madjid, Jaehong Yoon, Mark Hasegawa-Johnson, Sung Ju Hwang, and Chang D. Yoo. Forget-free continual learning with winning subnetworks. In *International Conference on Machine Learning*, 2022. 6

[13] Zixuan Ke, Bing Liu, and Xingchang Huang. Continual learning of a mixed sequence of similar and dissimilar tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 18493–18504, 2020. 4

[14] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. 3

[15] Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012. 6

[16] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017. 1, 3, 6

[17] David Lopez-Paz and Marc' Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, volume 30, pages 6467–6476, 2017. 1, 3, 6

[18] 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

[19] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018. 1

[20] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, volume 24, pages 109–165. Elsevier, 1989. 1

[21] Martial Mermillod, Aurélia Bugaiska, and Patrick BONIN. The stability-plasticity dilemma: investigating the continuum from catastrophic forgetting to age-limited learning effects. *Frontiers in Psychology*, 4, 2013. 1

[22] Zichen Miao, Ze Wang, Wei Chen, and Qiang Qiu. Continual learning with filter atom swapping. In *International Conference on Learning Representations*, 2022. 6

[23] Mozhgan PourKeshavarzi, Guoying Zhao, and Mohammad Sabokrou. Looking back on learned experiences for class/task incremental learning. In *International Conference on Learning Representations*, 2022. 6

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

[25] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016. 3

[26] Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *International Conference on Learning Representations*, 2021. 1, 3, 4, 6, 7

[27] Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning*, pages 4528–4537, 2018. 6

[28] Yujun Shi, Kuangqi Zhou, Jian Liang, Zihang Jiang, Jiashi Feng, Philip H.S. Torr, Song Bai, and Vincent Y. F. Tan. Mimicking the oracle: an initial phase decorrelation approach for class incremental learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 16722–16731, June 2022. 1

[29] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, volume 30, 2017. 1, 3, 6

[30] Kenneth W Spence and Janet Taylor Spence. *Psychology of learning and motivation*. Academic Press, 1967. 2, 3

[31] Stanford. Tiny imagenet challenge (cs231n). 2015. 6

[32] Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019. 2, 3

[33] Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training networks in null space of feature covariance for continual learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 184–193, 2021. 1, 3, 6

[34] Ju Xu and Zhanxing Zhu. Reinforced continual learning. In *Advances in Neural Information Processing Systems*, volume 31, pages 907–916, 2018. 3

[35] Hongxu Yin, Pavlo Molchanov, Jose M. Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K. Jha, and Jan Kautz. Dreaming to distill: data-free knowledge transfer via deepinversion. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2020. 3, 6

[36] Jaehong Yoon, Saehoon Kim, Eunho Yang, and Sung Ju Hwang. Scalable and order-robust continual learning with additive parameter decomposition. In *International Conference on Learning Representations*, 2020. 3

[37] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations*, 2018. 3

[38] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pages 3987–3995, 2017. 3, 6

[39] Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5871–5880, June 2021. 1, 3, 6