Do Your Best and Get Enough Rest for Continual Learning

Supplementary Material

1. Hyper-parameter Configuration

1.1. Baseline Method

We provide hyper-parameters for each dataset used in this paper. We describe our hyper-parameter settings for both from-scratch training and fine-tuning methods. We train ResNet-18 from scratch while employing a fine-tuning method for ViT. We use ViT-B/16, which is pre-trained on the ImageNet-21K dataset. Tabs. 1 and 2 show the key hyper-parameters used in our experiments. We follow other specific hyper-parameters by baseline methods' for a fair comparison with them. To provide more reliable experimental results, we report the mean and variance of three experimental results, each with different random seeds.

Table 1. **Hyper-parameters for ResNet-18 backbone.** We show the key hyper-parameters used in this paper.

Hyper-parameter	CIFAR-10	CIFAR-100	TinyImageNet	DomainNet
Epochs	50	170	100	50
Batch size	32	128	32	128
Optimizer	SGD	SGD	SGD	SGD
Learning rate	0.03 - 0. 1	0.03 - 0.1	0.03 - 0.1	0.03 - 0.1
LR scheduler	None	Multi-step	None	None
Weight decay	None	None	None	None
Network	ResNet-18			

Table 2. **Hyper-parameters for ViT-B/16 backbone.** We describe the hyper-parameter settings for pre-trained networks.

Hyper-parameter	CIFAR-100	ImageNet-R	DomainNet
Epochs	20	50	20
Batch size	128	128	128
Optimizer	SGD	SGD	SGD
Learning rate	0.01	0.01	0.01
Milestones	18	40	15
LR decay	0.1	0.1	0.1
Weight decay	5e-4	5e-4	5e-4
Network	ViT-B/16		

1.2. View-Batch Model

To ensure simplicity, we do not modify the baseline methods' hyper-parameters when applying our method to them. Therefore, our method does not require hyper-parameter changes from baseline methods and does not need extra training or inference costs compared to baseline methods. For augmentation type, we employ the widely-used auto-augmentation [2] method. We do not change augmentation types for different datasets or methods for strict comparison. This handcraft augmentation search will improve the network's performance, but we decided to stick to the same augmentation method to validate the proposed method's effect only without interference with augmentation.

Method	Latency	RAM_{CPU}	RAM_{GPU}: forward	RAM_{GPU}: backward
Baseline	29.5ms	1.431GB	0.210GB	0.235GB
+View-Batch-replay	27.5ms	1.422GB	0.210GB	0.235GB
+View-Batch-SSL	28.3ms	1.425GB	0.210GB	0.235GB

Table 3. **Experimental results on computing cost.** We use iCaRL as the baseline method on the S-CIFAR-10 dataset.

Method	**Task**	**RI=1**	**RI=2**	**RI=3**	**RI=4**	**RI=5**
iCaRL	CIL	64.11	65.85	68.39	69.73	67.67
	TIL	90.20	90.94	92.96	92.76	92.47
	Avg	77.16	78.39	80.68	81.25	80.07
DER++	CIL	61.67	65.79	66.99	61.68	63.44
	TIL	90.61	93.57	94.30	93.82	94.42
	Avg	76.14	79.68	80.65	77.75	78.93

Table 4. **Experimental results on recall interval.** We show the last top-1 accuracy varying the recall intervals (denoted as RI) on S-CIFAR-10. We use the ResNet-18 backbone in this experiment.

2. Forgetting Curve Analysis Backgrounds

In Section 1 of the manuscript, we illustrate the forgetting curve with different recall intervals. We draw these forgetting curves based on spacing effect theory [1, 3]. In spacing effect theory, we estimate memory retention according to elapsed time and recall interval. Specifically, in the forgetting curve theory [3], human's memory retention R could be defined as the function of the elapsed time t from the initial learning experience as:

$$R(t) = A(bt + 1)^{-S}, \tag{1}$$

where A is the first memory retention, b denotes time scaling parameters, and S represents the memory retention decay rate. Obviously, we assume a higher decay rate of memory retention indicates faster forgetting. Moreover, Cepeda et al. [1] empirically proves that the decay rate depends on recall interval I. Borrowing this empirical finding, the decay rate is defined as

$$S = 1 + c(ln(I + 1) - d)^2, \tag{2}$$

where d and c are empirically determined parameters. From Equation (2), we learn that the increasing recall interval improves the decay rate until some point d, then deteriorates

Method	5 Step				10 Step				20 Step			
	Avg	Δ	*Last*	Δ	*Avg*	Δ	*Last*	Δ	*Avg*	Δ	*Last*	Δ
DER	76.77	-	68.06	-	75.72	-	64.32	-	74.96	-	61.80	-
+replay	77.63	**+0.86**	69.21	**+1.25**	76.53	**+0.81**	65.49	**+1.17**	75.55	**+0.59**	62.65	**+0.85**
+self-supervised	78.60	**+1.83**	70.60	**+2.54**	78.12	**+2.40**	67.04	**+2.72**	76.95	**+1.99**	64.29	**+2.49**
TCIL	77.33	-	69.48	-	76.33	-	65.66	-	74.32	-	62.54	-
+replay	78.42	**+1.09**	70.35	**+0.87**	77.02	**+0.69**	67.71	**+2.05**	75.07	**+0.75**	63.93	**+1.39**
+self-supervised	79.23	**+1.90**	71.23	**+1.75**	78.02	**+1.69**	68.14	**+2.48**	76.83	**+2.51**	67.16	**+4.62**

Table 5. **Experimental results on factor analysis.** We showcase *Avg* and *Last* top-1 accuracy (%) on the S-CIFAR-100 benchmark with three different class incremental steps. ResNet-18 backbone is adopted for all networks. We follow the official implementation to reproduce the results of DER and TCIL. We demonstrate that our two main components significantly improve performance.

it again. Wahlheim et al. [5] interpret this phenomenon as optimal recall interval mitigates a high decay rate due to adequate learning difficulty. Finally, based on the given formula and theory, we illustrate different forgetting curves with varying recall intervals.

The degree of forgetting estimates the amount of memory neural networks forgets during recall intervals. Namely, if the degree of forgetting is high, the neural networks significantly lose their memory before they relearn the same samples. Not surprisingly, since excessive memory forgetting yields poor long-term memory retention in human learners [4], we employ our degree of forgetting as an empirical reason for the downward accuracy phenomenon in long-term recall interval in Figure 4 of the manuscript.

Specifically, we define the sequence of memory retention as $r_0, r_1, ..., r_{E-1}$. Here, we denote E for the number of total learning epochs and measure r_i by evaluating neural networks on the current task at the end of each epoch and adopting their top-1 accuracy (%) as a memory retention value. Since we aim to quantify the degree of forgetting during recall interval, the variance of the memory retention values is used as our metric, calculating the averaged memory retention differences between the mean and individual retention values. Leveraging the average memory retention difference, we define our degree of forgetting Δ_r.

$$\Delta_r = \frac{1}{E-\overline{E}} \sum_{i=\overline{E}}^{E} ((\frac{1}{E-\overline{E}} \sum_{j=\overline{E}}^{E} r_j) - r_i)^2, \qquad (3)$$

where we include memory retention values from the saturated epochs $\overline{E}$ due to high randomness in the early learning phase. Consequently, we could evaluate the degree of forgetting of neural networks based on Equation (3) in various CL scenarios.

3. Additional Experimental Results

3.1. Results on Training Cost

Tab. 3 analyzes the resources overhead in terms of latency (ms) and CPU and GPU RAM usage, which is measured by averaging three runs. The proposed method increases the minimal latency by 3% to compute the KL divergence loss. Further, there is no additional resource usage for both CPU and GPU RAM when using our method. Therefore, we demonstrate that the proposed method can be utilized as the drop-in replacement approach.

Method	Degree of forgetting	Avg top-1 accuracy (%)
Baseline	**1.73**	76.33
VBM-C	**6.20**	74.68
VBM-S	**2.73**	78.11

Table 6. **Experimental results on View-Batch target.** We show the degree of forgetting and its average top-1 accuracy (%) on the S-CIFAR-100 dataset varying the targets of view-batch model. For the degree of forgetting, we represent prohibitive values with **red** and mark optimal values with **green** color.

3.2. Results on Recall Interval

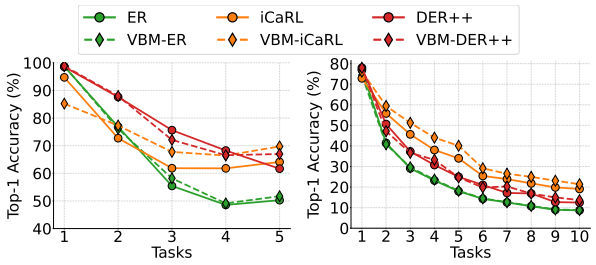
Tab. 4 validates the various recall intervals on the S-CIFAR-10 datasets using two different baseline methods. We show that the optimal recall interval (*i.e.*, **x3** or **x4**), which is found in Section 6 of the manuscript, generally works well in different baseline methods and different task types. This analysis demonstrates the proposed method's applicability in various continual learning scenarios.

3.3. Results on Factor Analysis

We extensively perform factor analysis in different CL scenarios. Tab. 5 demonstrate that the main components significantly improve respective baseline methods. These experimental results reveal that the proposed components work consistently well across diverse CL scenarios.

3.4. Results on two types of View-Batch Model

We compare two types of the view-batch model, such as class- and sample-based approaches. While we augment a single sample to multiple views for constructing a view-batch, one can also do this view-batch construction in a



(a) S-CIFAR-10 (b) S-TinyImageNet

Figure 1. **Experimental results on evolving task.** We report class incremental accuracy at the end of each task, comparing the view-batch model to baseline methods. We provide details of the task evolution results in our repository.

class-based manner. For the class-based view-batch (VBM-C) method, we constrain a single epoch to learn only specific class samples. If there are C classes, we train only C/N class samples N repeated times per epoch. This allows networks to learn the features of a specific class extensively in a single epoch. However, Tab. 6 shows that the VBM-C over-escalates the degree of forgetting, leading to lower performance of continual learning. On the other hand, our sample-based approach (VBM-S) increases memory retention fairly and achieves favorable performance compared to the class-based one. We assume that our sample-based approach balances the recall interval and extensive learning. Moreover, it indicates that it is more advantageous for self-supervised learning to learn all class samples in a single epoch.

3.5. Results on Task Evolution

We evaluate task evolution performances in the S-CIFAR-100 dataset as shown in Figure 1. In task evolution, we measure average class incremental accuracy at the end of each task and report all of them to compare our view-batch model and its respective baseline. As a result, our view-batch model shows consistent performance improvements in all steps of various evaluation scenarios.

References

[1] Nicholas J Cepeda, Edward Vul, Doug Rohrer, John T Wixted, and Harold Pashler. Spacing effects in learning: A temporal ridgeline of optimal retention. *Psychological science*, 2008. 1

[2] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv:1805.09501*, 2018. 1

[3] Hermann Ebbinghaus. Memory: A contribution to experimental psychology. *Annals of neurosciences*, 2013. 1

[4] Arthur W Melton. The situation with respect to the spacing of repetitions and memory. *Journal of Verbal Learning and Verbal Behavior*, 1970. 2

[5] Christopher N Wahlheim, Geoffrey B Maddox, and Larry L Jacoby. The role of reminding in the effects of spaced repetitions on cued recall: sufficient but not necessary. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 2014. 2