Do Pre-trained Models Benefit Equally in Continual Learning? Supplementary Material

Kuan-Ying Lee Yuanyi Zhong Yu-Xiong Wang

University of Illinois at Urbana-Champaign

{kylee5, yuanyiz2, yxw}@illinois.edu

A. Additional Implementation Details

Detailed Setting of Each Experiment. All experiments in the main paper are performed on Split CIFAR100 in online CIL, if not specified otherwise. Table A elaborates the details of each table in the main paper. As discussed in the Introduction, we attempt to fix one axis and analyze the effect of the other two. For example, in Table 2, we fix the CL scenario and analyze the benefits of pre-trained models. In Table 5, we fix the CL algorithm and compare pre-trained models and CL scenarios.

Metrics. Accuracy refers to the all-way classification accuracy at the end of the training, e.g., 100-way classification on Split CIFAR100. **Forgetting** is computed by subtracting the accuracy of a task right after it is learned by the final accuracy of the same task.

B. Two-Stage Training Pipeline

Methodology. The two-stage training is dependent on the underlying CL algorithm. We regard the memory as the training set and apply the exact CL algorithms on the "memory dataset," but in an i.i.d. fashion for multiple (30) epochs. See Fig. A for the pipeline diagram.

Comprehensive Results on Different CL Algorithms. As discussed in Table 7 of the main paper, the two-stage training pipeline that combines learning in the streaming phase and offline training with samples in the memory could make a simple ER method a strong baseline when coupled with a pre-trained model (ImageNet RN50). In Table B, we further apply the two-stage training to the other CL algorithms and additionally compare the two-stage training between initialization from scratch (R-RN18+TS) and a pre-trained model (RN18+TS and RN50+TS).

Three observations are made here. 1) The absolute improvement brought by the two-stage training pipeline is more pronounced with an ImageNet RN (RN18+TS vs. RN18 and RN50+TS vs. RN50) compared with from-scratch training (R-RN18+TS vs. R-RN18). 2) The two-stage training pipeline is generally beneficial across CL algorithms, despite its slightly negative impact on SCR. 3) **ER shows the best performance** despite its simplicity, when coupled with ImageNet RN50 and the two-stage training pipeline. This clearly shows the significance of our observations (minimum regularization benefits more from a pre-trained model; ImageNet RN50 generally outperforms CLIP RN50), which facilitates the proposed strong baseline.

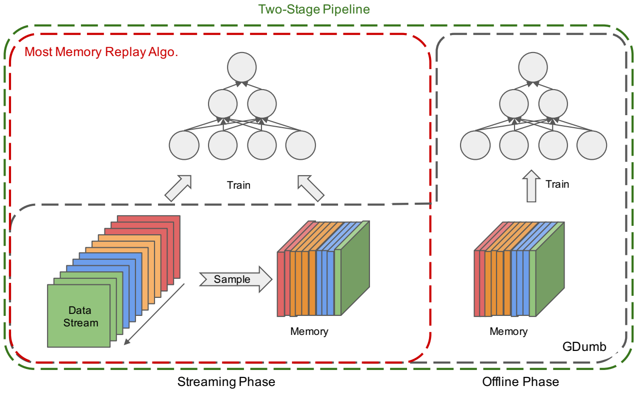

Figure A. **Two-stage training CL pipeline.** Most memory replay methods only perform learning during the streaming phase while on the contrary, GDumb only performs learning at the end of the stream (offline phase). Coupled with a pre-trained model, this simple two-stage pipeline that learns in both phases converts Experience Replay (ER) into a state-of-the-art approach (cf. Table B).

C. Discussion on Forgetting

Less Forgetting with CLIP RN50. In Table 4 of the main paper, we discussed that CLIP shows less forgetting, compared with ResNets trained with the ImageNet data in a supervised manner. We conjecture the lower forgetting might be attributed to 1) the lower learning rate required for CLIP

Main paper	Models	CL Algorithm	CL Scenario
Table 2	R-RN18, RN18	All	Online CIL
Table 3	All	All	Online CIL
Table 4	All	All but DER++, Co^{2}L	Online CIL
Table 5	R-RN18, RN18, RN50, SimCLR RN50	ER, LwF	All
Table 6	RN50, CLIP	ER	Online CIL
Table 7	RN18, RN50, CLIP	ER, iCaRL, SCR	Online CIL
Fig. 1	RN18	All but DER++, Co^{2}L	Online CIL
Fig. 2	R-RN18, RN18, RN50, CLIP, SimCLR	ER	All
Fig. 3	RN50, CLIP	ER	Online CIL

Table A. **Detailed configurations** of experiments in the main paper. We attempt to analyze pre-trained models in CL through these three axes – different pre-trained models, different CL algorithms, and different CL scenarios.

Model	ER [11]	MIR [1]	GSS [2]	*LwF* [7]	iCaRL [10]	*EWC++* [4]	GDumb [9]	AGEM [5]	SCR [8]
R-RN18	9.07±1.31	8.03±0.78	6.86±0.60	8.44±0.82	14.26±0.79	1.00±0.00	9.80±0.46	3.00±0.47	**25.80±0.99**
+TS	14.66±0.23	13.67±0.47	12.67±0.25	—*	14.66±0.35	—*	—†	12.50±0.58	**23.06±0.09**
Δ	5.59	5.64	5.81	—*	0.40	—*	—†	**9.50**	-2.74
RN18	43.69±1.67	42.02±1.53	25.59±0.45	23.40±0.12	**56.64±0.23**	5.36±0.26	46.76±0.73	4.72±0.21	51.93±0.06
+TS	58.59±0.31	56.64±0.14	37.08±1.45	—*	58.66±1.07	—*	—†	57.98±0.57	49.55±0.29
Δ	14.90	14.62	11.49	—*	2.02	—*	—†	**53.26**	-2.38
RN50	50.88±0.84	50.20±2.80	31.53±3.37	26.68±0.97	**59.20±0.33**	3.47±1.42	57.37±0.21	4.49±0.27	56.22±0.42
+TS	**65.35±0.55**	62.87±0.63	52.03±0.62⋆	—*	60.44±0.13	—*	—†	62.76±0.54	51.55±0.24
Δ	14.47	12.67	20.5	—*	1.24	—*	—†	**58.27**	-5.92

*We do not apply the two-stage training to LwF and EWC++, because no memory buffer is employed.

†GDumb is essentially the second stage;

Table B. **Two-stage accuracy** on Split CIFAR100 in online CIL. While the two-stage training pipeline is generally beneficial, a performance drop is present in SCR (-2.38). ER shows the best performance despite its simplicity, when coupled with ImageNet RN50 and the two-stage training pipeline. The green numbers indicate a positive accuracy increase while the red numbers indicate a decrease, when the two-stage training pipeline is applied. **Bold** numbers indicate the best accuracy amongst all methods with a specific setting (e.g., 25.80 of SCR is the best with R-RN18). R-RN18 and RN18 stand for Reduced ResNet18 trained from scratch and ImageNet pre-trained ResNet18, respectively. **[Best viewed in color.]**

to train successfully, and 2) the feature normalization that projects all features on a unit hyper-sphere, potentially serving as a form of regularization.

Less Forgetting with Self-supervised Fine-tuning. In Table 5 of the main paper, we observed that self-supervised fine-tuning (with the SimCLR loss) shows less forgetting compared with supervised counterparts in the downstream task. Here, we attempt a more in-depth examination.

Self-supervised fine-tuning of SimCLR involves 1) self-supervised update of features and 2) supervised training of the classifier. The difference between SimCLR and fine-tuning networks in a supervised manner is two-fold: 1) the decoupling of feature and classifier training, and 2) the features are learned in a self-supervised fashion. We attempt to isolate the effect of the decoupling mechanism. To achieve this, we train RN18 via the usual supervised cross-entropy loss with the ER method. However, at the end of each task, we discard the learned classifier and instead train a new one from scratch with the samples in ER (as we do when fine-tuning SimCLR RN50 in a self-supervised manner). Doing so, the forgetting is now 29.13±0.97, which is significantly lower than its counterpart ImageNet RN50 (42.93±0.67 as in Table 4 of the main paper) whose feature and classifier are trained jointly.

This potentially indicates that only a minor part of the forgetting results from the supervised training of the features and it is the joint training of the features and the classifier that causes the most forgetting. Such an observation might be loosely connected to [6], where it was found harmless to train features with long-tailed distributed data. However, when training the classifier, a re-balanced dataset is instead utilized. This is, to some degree, analogous to CL with a replay buffer since during the streaming phase, most data come from the current task (set of classes) while previous classes only account for a small portion (fairly assuming the replay is relatively small compared with the size of data from the current task). In other words, similarly, during the streaming phase in CL, one conducts feature learning with

Model	Co^{2}L [3]	Co^{2}L + Two-Stage
R-RN18	2.31±0.64	6.30±0.46
RN18	5.68±3.19	38.71±1.05
RN50	8.57±0.57	43.80±0.73
CLIP	1.12±0.16	17.93±1.21
SimCLR	1.44±0.45	14.86±0.29
SwAV	1.18±0.26	9.45±0.61
Barlow Twins	1.10±0.10	15.26±0.67

Table C. **Co^{2}L with two-Stage training.** Accuracy increases largely when the classifier is trained for 30 epochs instead of just one.

a quasi-long-tailed distribution.

Connection to Two-Stage Pipeline. The aforementioned experiment potentially explains why the two-stage mechanism further improves the performance. Since training the classifier jointly during the streaming phase shows the most forgetting, the second (offline) stage mitigates the forgetting by learning the classifier with a balanced sample set (samples in the memory). This could also be analogous to the second step of [6], training the classifier with a balanced sample set.

D. Co^{2}L with Two-Stage Training

In Table C, after Co^{2}L learns the feature in the streaming phase, instead of training the classifier for one epoch, we train it for 30 epochs. By doing so, accuracy increases significantly.

References

[1] R. Aljundi, E. Belilovsky, T. Tuytelaars, L. Charlin, M. Caccia, M. Lin, and L. Page-Caccia. Online continual learning with maximally interfered retrieval. In *NeurIPS*, 2019. 2

[2] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio. Gradient based sample selection for online continual learning. In *NeurIPS*, 2019. 2

[3] H. Cha, J. Lee, and J. Shin. Co2L: Contrastive continual learning. In *ICCV*, 2021. 3

[4] A. Chaudhry, P. K Dokania, T. Ajanthan, and P. HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *ECCV*, 2018. 2

[5] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny. Efficient lifelong learning with a-gem. In *ICLR*, 2019. 2

[6] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *ICLR*, 2020. 2, 3

[7] Z. Li and D. Hoiem. Learning without forgetting. *TPAMI*, 40(12):2935–2947, 2017. 2

[8] Z. Mai, R. Li, H. Kim, and S. Sanner. Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning. In *CVPR*, 2021. 2

[9] A. Prabhu, P. HS Torr, and Puneet K. D. GDumb: A simple approach that questions our progress in continual learning. In *ECCV*, 2020. 2

[10] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. iCaRL: Incremental classifier and representation learning. In *CVPR*, 2017. 2

[11] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne. Experience replay for continual learning. In *NeurIPS*, 2019. 2