Supplementary Materials for Revisiting Supervision for Continual Representation Learning

Daniel Marczak*[1,2], Sebastian Cygert[1,3], Tomasz Trzciński[1,2,4], and Bartłomiej Twardowski[1,5,6]

[1] IDEAS NCBR
[2] Warsaw University of Technology
[3] Gdańsk University of Technology
[4] Tooploox
[5] Autonomous University of Barcelona
[6] Computer Vision Center

A Implementation details

A.1 CL strategies

LwF [4] is a classic SCL method for feature distillation. It distills the logits of the frozen network trained on previous tasks using cross-entropy loss. We pair it with SL methods that train with cross-entropy loss. We use the implementation from [5].

CaSSLe [2] is a method for self-supervised continual learning that utilizes a learnable MLP to project past features onto the new latent space for improved feature distillation. The distillation is performed on the outputs from the SSL projector with the loss function of a particular SSL method. Because of reliance on SSL-specific components, namely the projector and loss function, we do not pair CaSSLe with supervised approaches, except for SupCon which loss and architecture closely resemble SSL. We follow an official implementation of CaSSLe.

PFR [3] realizes a similar idea to CaSSLe. It also uses a learnable MLP projector to enhance feature distillation. However, it uses cosine similarity as a loss function and performs distillation on the outputs of the backbone network. Therefore, we pair it with both SL and SSL approaches as it does not rely on SSL-specific components. We present the chosen values of regularization hyperparameter λ in Table 1. We selected the best $\lambda \in \{1, 3, 10, 15, 25\}$ separately for each method and dataset.

A.2 k-NN evaluation

Each model is evaluated with a k-nearest neighbour classifier after training each task (offline evaluation). Moreover, we perform some experiments where we use k-nn evaluation after each epoch (online evaluation for Figure 4).

For online evaluation, we perform extensive hyperparameter search and report results obtained by the best probe. We explore the following hyperparameters:

* Corresponding author, email: daniel.marczak.dokt@pw.edu.pl

Table 1: PFR regularization hyperparameter λ for different methods and datasets.

Method	C10/5	C100/5	C100/20	IN100/5
SL	1.0	10.0	10.0	15.0
SL+MLP	3.0	3.0	10.0	1.0
t-ReX	3.0	3.0	10.0	1.0
SupCon	3.0	10.0	25.0	10.0
BarlowTwins	25.0	25.0	25.0	25.0
SimCLR	3.0	3.0	15.0	3.0

- $k \in \{5, 10, 20, 50, 100, 200\}$ - number of considered neighbours;
- distance function - we consider either euclidean distance or cosine similarity;
- temperature $T \in \{0.02, 0.05, 0.07, 0.1, 0.2, 0.5\}$ used only with cosine distance;

resulting in 42 k-NN probes per one offline evaluation.

For online evaluation, we use a fixed hyperparameter set: $k = 20$, cosine distance, and $T = 0.07$. This k-NN configuration often turns out to be one of the best in offline evaluation.

A.3 Architectures of the projector

In Figure 1 we present the architectures of the projectors proposed by SimCLR [1], t-ReX [6] and by [7]. The results of different methods paired with these projectors are presented in Table 6 in the main paper.

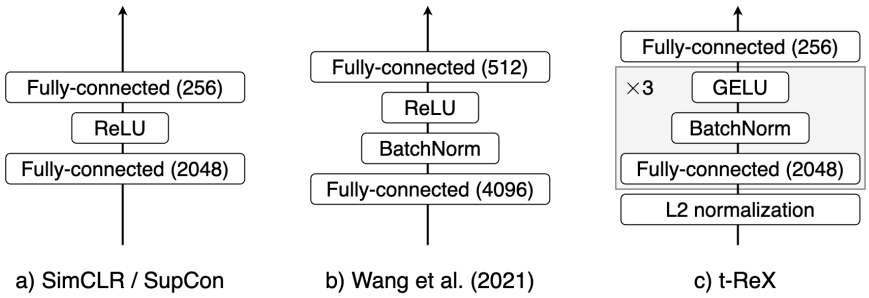

Figure 1: Architectures of the projectors used by different methods.

B Extended analysis

B.1 Initialization of MLP projector

Is it better to randomly reinitialize the projector after each task or is it better to start from the weights of the projector learned on the previous tasks? The results

reported in Table 2 suggest that for SL it is better to randomly reinitialize the MLP projector. SSL methods, however, tend to perform slightly better when the projector for a new task is initialized with the previously learned projector. We suspect that in SL the projector encompasses task-specific knowledge which interferes with learning new tasks. On the other hand, in SSL the MLP is responsible for projecting the representations into the space where an invariance to augmentations is enforced which is less task-specific than classification.

Table 2: SL benefits from resetting the MLP projector and SSL methods tend to perform slightly better when starting from the weights of the projector learned on previous tasks. We report k-NN accuracy (%) after the final task. Better initialization method in **bold**.

Method	MLP init	CIFAR10/5	CIFAR100/5
SL+MLP	Reset	**65.9±0.7**	**61.9±0.5**
	Previous	65.0±1.5	60.1±0.2
BarlowTwins	Reset	76.1±0.5	**54.9±0.3**
	Previous	**76.2±1.2**	54.1±0.3
SimCLR	Reset	72.0±1.6	47.4±0.2
	Previous	**72.4±1.3**	**48.9±0.4**

B.2 Stability of representations

We define representations as *stable* when they do not drift in the representation space when the network is trained on a new task. The stability of representations is a desired property of SCL models as stable representations facilitate continual training of a classifier [8]. On the other hand, UCL evaluation only measures the representations' strength and the relationship of stability and strength of representations is not obvious. One can imagine both stable and unstable representations can improve strength during continual training.

In this section, we evaluate the stability of representations of SL and SSL models. We use nearest mean classifier (NMC) accuracy to measure it in the context of SCL. After the first task, we calculate prototypes of each class as a mean feature of all the samples of this class. We evaluate the model and save the prototypes. Then, we train on the second task and evaluate the model using saved prototypes. We use the accuracy obtained by classification using old prototypes as a proxy of the stability of the representations. In the case of perfectly stable representations, both evaluations would result in the same accuracy while perfectly unstable representations would cause accuracy to drop to a random guess level. Moreover, we evaluate the updated model using prototypes recalculated on previous data (not allowed in continual learning) to provide an upper bound.

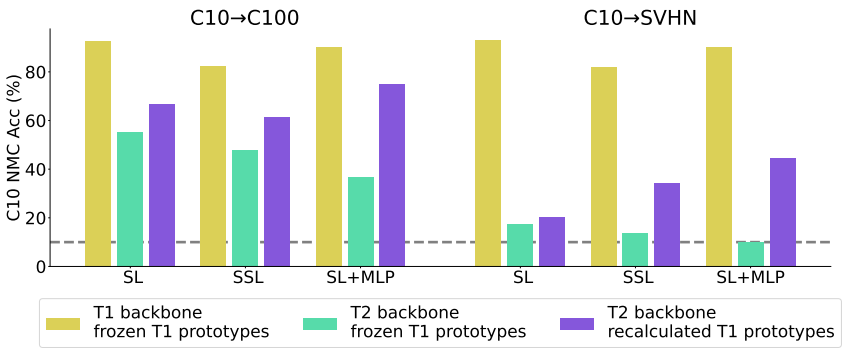

Figure 2: Task aware NMC accuracy on CIFAR10 dataset for supervised and self-supervised models trained on different sequences of tasks. After training on CIFAR10 (T1), both SL and SSL models achieve high NMC performance (yellow). After training the second task (T2), the nearest mean classification using old prototypes results in performance degradation (green). We calculate *an upper-bound* accuracy after training on the second task by recalculating the prototypes using old data and a new backbone (purple). Note that it is not possible in the CL scenario as old data is inaccessible. Gray dotted line marks random guess performance.

The results are presented in Figure 2. Representations of all the methods are not stable in high distribution shift scenario C10→SVHN. They achieve random guess accuracy when utilizing saved (old) prototypes. However, in a low distribution shift scenario, C10→C100, SL achieves 55.3% accuracy using old prototypes (11.3% below upper bound performance) while SSL achieves 48.0% (14.5% below upper-bound) and SL+MLP achieves only 36.7% (38.3% below upper-bound). Note that performance degradation can be only partially attributed to forgetting of representations as the upper-bound performance is still high after training on the second task for most of the methods. These results suggest that there exists a trade-off between the stability of representations and the expressiveness of representations trained continually as methods that build stronger representations tend to have lower stability.

B.3 Impact of training length

We investigate how the number of epochs influences the representations trained with different methods. We conducted experiments on a long sequence of tasks, C100/20, training with SSL, SL, and SL+MLP methods for different numbers of epochs in each task. We present the results in Table 3. We observe that a large number of epochs (500) is important for SSL to achieve good final results. However, the performance gap between the SSL model trained on 500 epochs and the SSL models trained for 100 or 200 epochs is decreasing with a number of tasks.

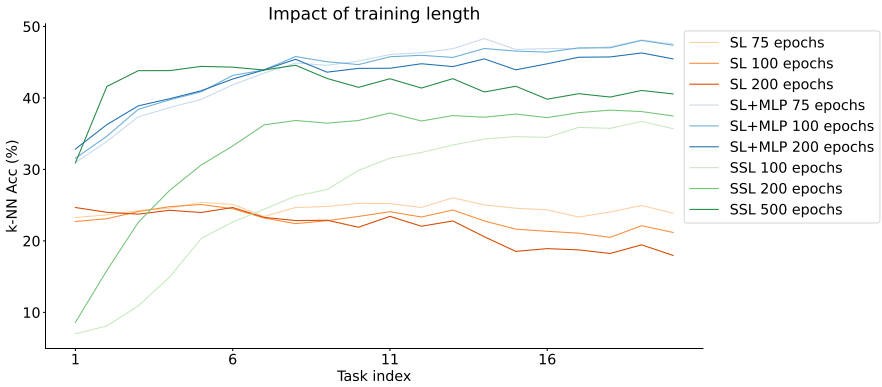

Figure 3: Self-supervised models benefit from longer training. However, supervised models, both with and without MLP projector result in reduced performance when trained for a large number of epochs. We report task-agnostic k-NN accuracy for all tasks after each task.

B.4 Task exclusion comparison

In Figure 4 we take a closer look at the task exclusion comparison. We identify that the training recipe is a factor responsible for its negative task exclusion difference. The training recipe for SL and SSL differs: SL is trained for 100 epochs with a 0.025 learning rate while SSL is trained for 500 epochs with 0.3 learning rate. When training SSL for 100 epochs with a learning rate of 0.025, following the SL+MLP learning recipe, we observe that SSL exhibits positive behavior that is similar to SL+MLP. However, such training configuration leads to the suboptimal final performance of a continual learner, as shown in Figure 3.

B.5 Detailed two-task results

In Figure 5 we present detailed results of two-task settings results summed up in Figure 1 in the main paper: C10→C100, C100→C10, C10→SVHN, SVHN→C10, C100→SVHN and SVHN→C100. We can observe that self-supervised learning usually outperforms supervised learning on the first task. The opposite is true for the second task – SL performs better than SSL. However, SL equipped with MLP achieves the highest average accuracy on both tasks usually outperforming both SL and SSL on the first and second tasks.

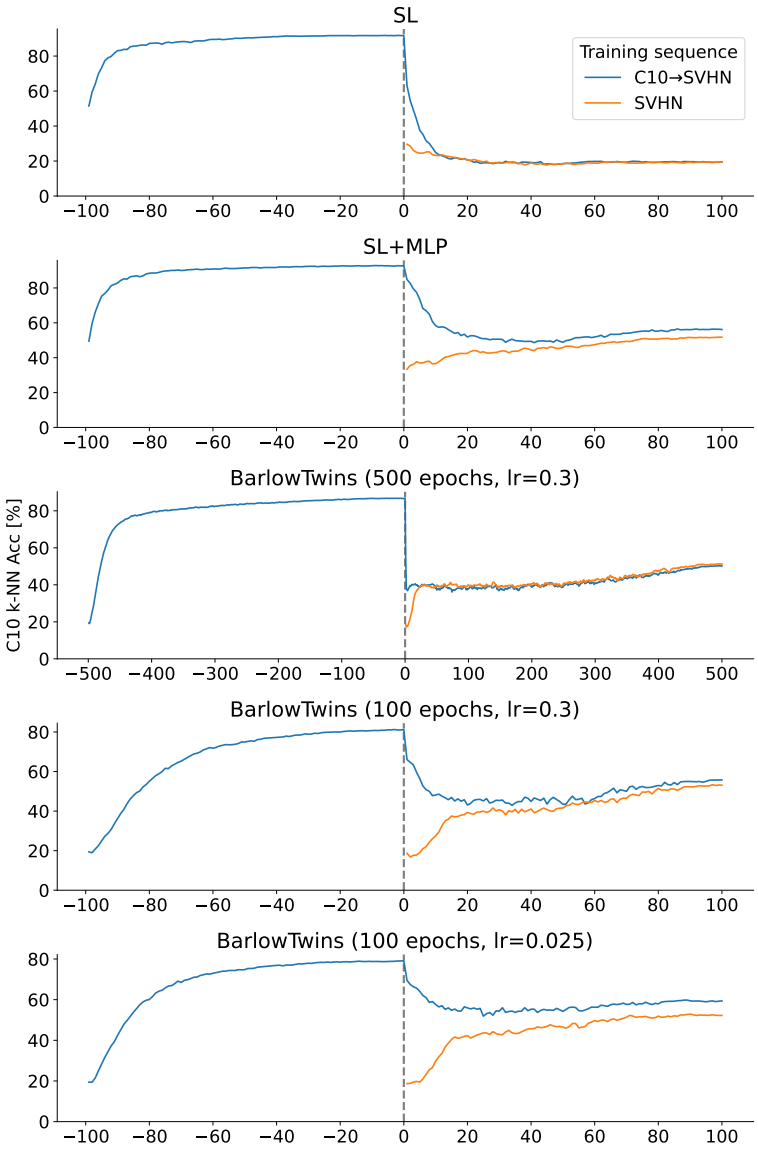

Figure 4: SSL behaves similarly to SL+MLP when trained for the same number of epochs with the same learning rate.

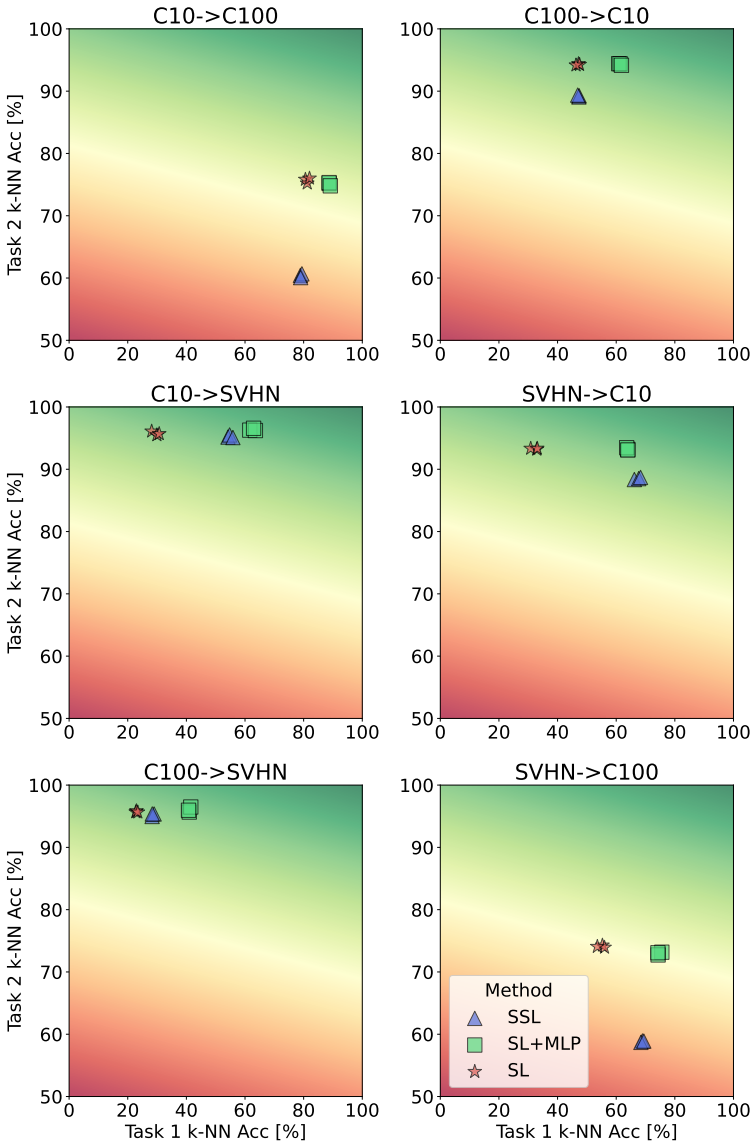

Figure 5: Results of two-task settings after training on the second task. Accuracy on the first task is presented on the horizontal axis and accuracy on the second task is presented on the vertical axis while the background color indicates the average accuracy on both tasks. SL usually outperforms SSL on the second task and usually underperforms on the first task. SL+MLP takes the best of both worlds (high first-task accuracy from SSL and high second-task accuracy from SL) and achieves the best overall performance.

References

1. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.E.: A simple framework for contrastive learning of visual representations. In: ICML (2020)
2. Fini, E., da Costa, V.G.T., Alameda-Pineda, X., Ricci, E., Alahari, K., Mairal, J.: Self-supervised models are continual learners. In: CVPR (2022)
3. Gomez-Villa, A., Twardowski, B., Yu, L., Bagdanov, A.D., van de Weijer, J.: Continually learning self-supervised representations with projected functional regularization. CVPR Workshops (2021)
4. Li, Z., Hoiem, D.: Learning without forgetting. IEEE TPAMI (2018)
5. Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A.D., van de Weijer, J.: Class-incremental learning: Survey and performance evaluation on image classification. IEEE TPAMI (2023)
6. Sariyildiz, M.B., Kalantidis, Y., Alahari, K., Larlus, D.: No reason for no supervision: Improved generalization in supervised models. In: ICLR (2023)
7. Wang, Y., Tang, S., Zhu, F., Bai, L., Zhao, R., Qi, D., Ouyang, W.: Revisiting the transferability of supervised pretraining: an mlp perspective. CVPR (2021)
8. Yu, L., Twardowski, B., Liu, X., Herranz, L., Wang, K., Cheng, Y., Jui, S., van de Weijer, J.: Semantic drift compensation for class-incremental learning. In: CVPR (2020)