Calibration of Continual Learning Models

Supplementary Material

Table 3. Best hyper-parameters for the continual training phase.

	S. MNIST	*S. CIFAR100*	*EuroSAT*	*Atari*
lr	1e-3	1e-2	1e-3	5e-4
mb size	32	128	128	256
epochs	20	50	20	100
validation split	0.2	0.2	0.1	0.2
patience	–	–	–	10
memory size	2000	4000	2000	4000
DER++				
α	0.3	0.2	0.1	0.5
β	0.8	0.8	0.5	0.5

Table 4. Best hyperparameters for the calibration approaches.

	S. MNIST	*S. CIFAR100*	*EuroSAT*	*Atari*
Joint				
ST - λ	0.0075	0.025	0.0075	0.0075
TS - lr	0.01	0.01	0.01	0.001
VS - lr	0.01	0.01	0.01	0.001
MS - lr	0.01	0.01	0.01	0.01
DER++				
HR - λ	0.005	0.025	0.025	0.025
TS - lr	0.01	0.01	0.01	0.01
VS - lr	0.01	0.01	0.01	0.01
MS - lr	0.01	0.01	0.01	0.01
Replay				
HR - λ	0.025	0.025	0.025	0.0025
TS - lr	0.01	0.01	0.01	0.01
VS - lr	0.01	0.01	0.01	0.001
MS - lr	0.01	0.01	0.01	0.001
Naive				
HR - λ	0.075	0.1	0.0075	0.005
TS - lr	0.01	0.01	0.01	0.001
VS - lr	0.01	0.01	0.01	0.001
MS - lr	0.01	0.01	0.01	0.001

6. Experimental setup

We report the optimal hyperparameters for all the benchmarks, different CL strategies and calibration techniques. Table 3 provides the best hyper-parameters for the CL training. Table 4 provides the same information about calibration techniques.
In Split MNIST and Atari we use SGD and Adam respectively with default values. In Split CIFAR100 and EuroSAT the chosen optimizer is AdamW (*weight_decay* $= 0.0005$) and we adopt as learning rate scheduler Cosine Annealing with Warm Restarts (*first restart iteration* $= 5$, *minimum lr* $= 0.00001$). For all the post-processing calibration techniques we fixed the number of training iterations to 100.

7. DER++ implementation

We used the DER++ version present in Avalanche [5]. Since the experimental setup and the details of the implementation may differ between the original version [4] and the Avalanche version, we ran some experiments to compare the performance. Table 5 shows that the average test accuracy on Split CIFAR100 and Split TinyImageNet obtained at the end of training does not change.
We used this sanity-check to ensure that the calibration performance of DER++ does not depend on a custom DER++ version.

8. Sensitivity of MS to changes in the learning rate

Figure 7 shows that calibration with MS on Joint Training is very sensitive to the choice of the learning rate. The ECE jumps from 10% to 60%, depending on the chosen learning rate.

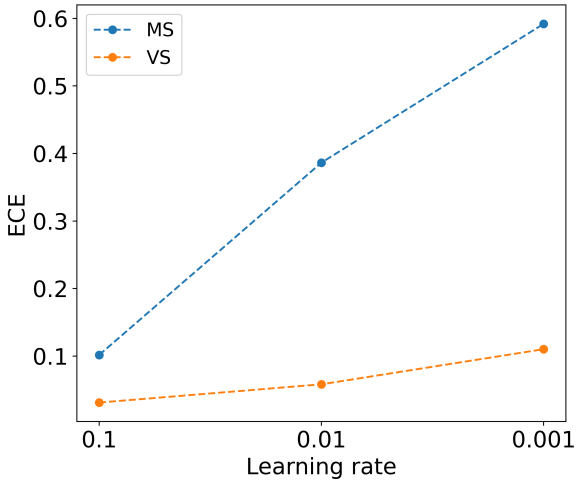

Figure 7. Sensitivity of Joint Training and MS to the learning rate. The dataset is Split CIFAR 100. The ECE does not change much for VS, while MS shows a large sensitivity on the chosen learning rate.

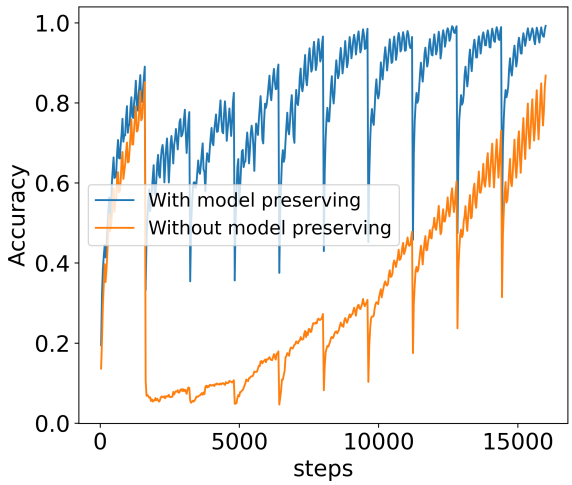

Figure 8. Comparison between re-training and discarding the wrapped model after the calibration phase.

9. Decision between retaining or discarding the wrapped model after calibration

In the post-processing Calibration method, we adjust the softmax temperature after the output layer or introduce additional linear projection during the calibration phase through temperature scaling or vector/matrix scaling. In CL scenarios, we encounter a sequence of experiences, where each experience concludes with a calibration phase following training. This alternation between training and calibration phases presents the option to either retain the wrapped model after the calibration phase or discard it for each training phase, utilizing it exclusively during calibration. We conduct experiments to explore both approaches. Figure 8 demonstrates a scenario involving training the DER model with Adamw + replayed matrix scaling calibration on the Cifar100 dataset. Here, we compare the accuracy between retaining and discarding the wrapped calibration model after the calibration phase. Notably, discarding the wrapped model results in complete forgetting after the first experience, necessitating the model to essentially "re-learn" as depicted in the figure. Conversely, retaining the wrapped model showcases a more stable learning curve, yielding higher accuracy and lower ECE. Based on these experimental observations, we choose to preserve the wrapped model after each calibration phase for all post-processing calibration experiments.

10. Reliability diagrams

We report the complete set of reliability diagrams for each benchmark and strategy.

Table 5. Comparison between the published results from DER++ and results obtained with our implementation of DER++ on Split CIFAR100 and Split Tiny ImageNet. We successfully replicate the results from the original papers.

Accuracy (%)	***S. CIFAR100*** [3]	***S. TinyImageNet*** [4]
Joint	70.44	59.99 ± 0.19
DER++	53.63	10.96 ± 1.17
Replay	38.58	8.49 ± 0.16
Naive	9.43	7.92 ± 0.26
Joint (*ours*)	69.00 ± 4.96	62.00 ± 0.52
DER++ (*ours*)	51.91 ± 0.93	12.83 ± 0.30
Replay (*ours*)	40.47 ± 0.95	10.10 ± 0.28
Naive (*ours*)	9.07 ± 0.10	7.52 ± 0.04

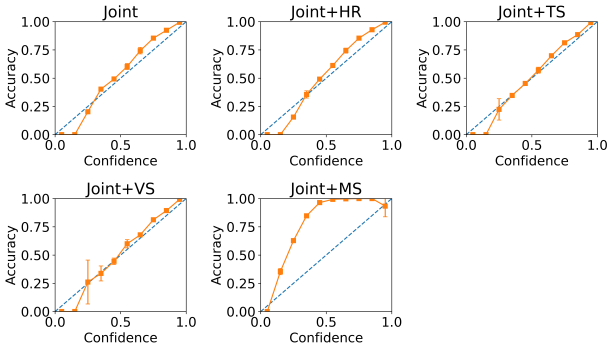

Figure 9. Reliability diagrams for Joint on Split MNIST

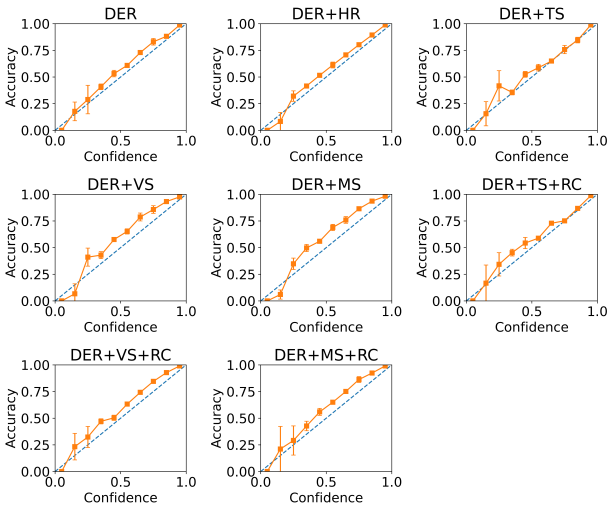

Figure 10. Reliability diagrams for DER++ on Split MNIST

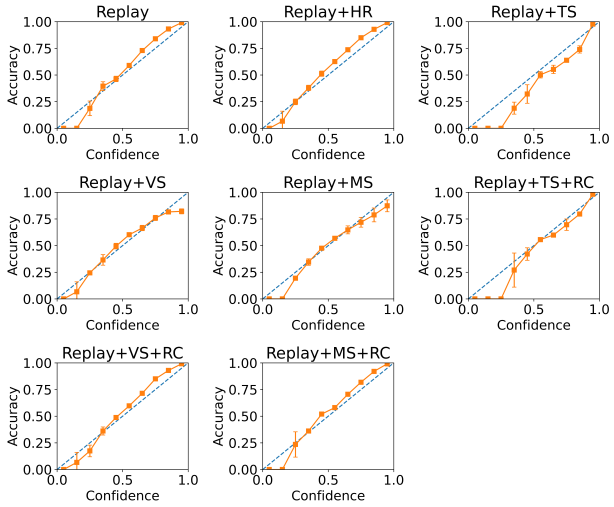

Figure 11. Reliability diagrams for Replay on Split MNIST

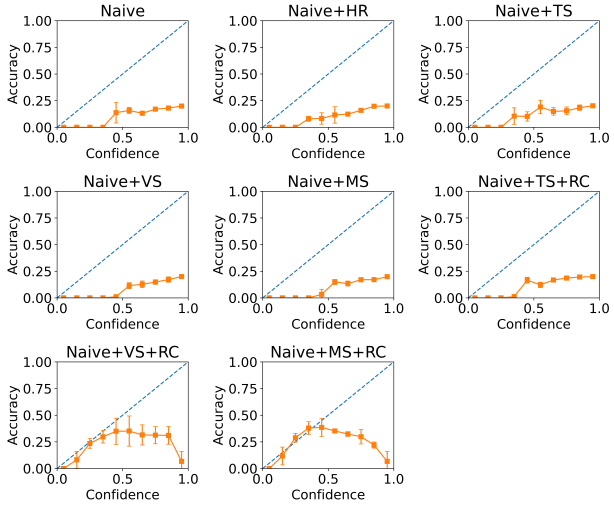

Figure 12. Reliability diagrams for Naive on Split MNIST

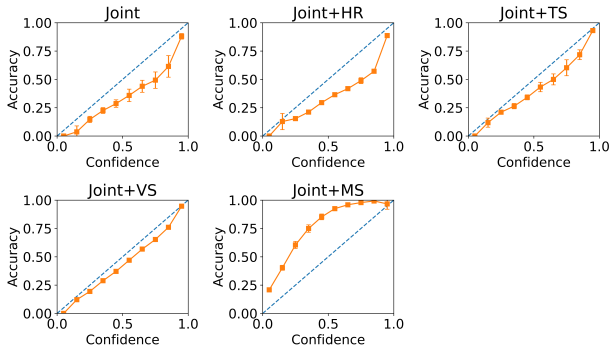

Figure 13. Reliability diagrams for Joint on Split CIFAR100

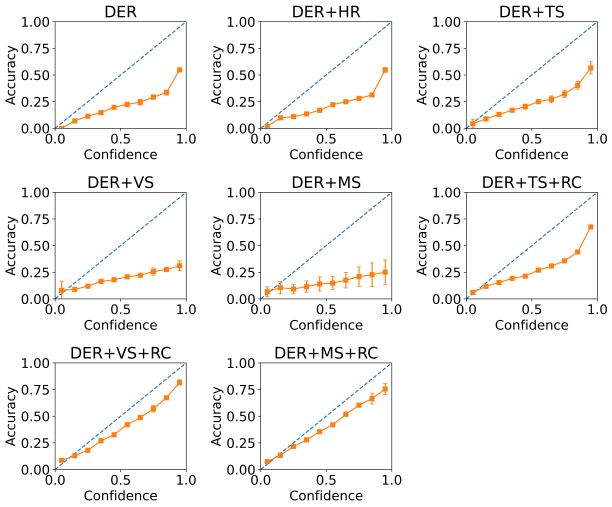

Figure 14. Reliability diagrams for DER++ on Split CIFAR100

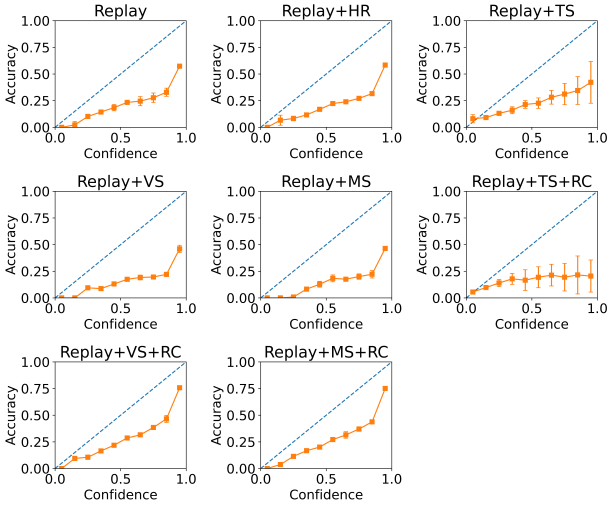

Figure 15. Reliability diagrams for Replay on Split CIFAR100

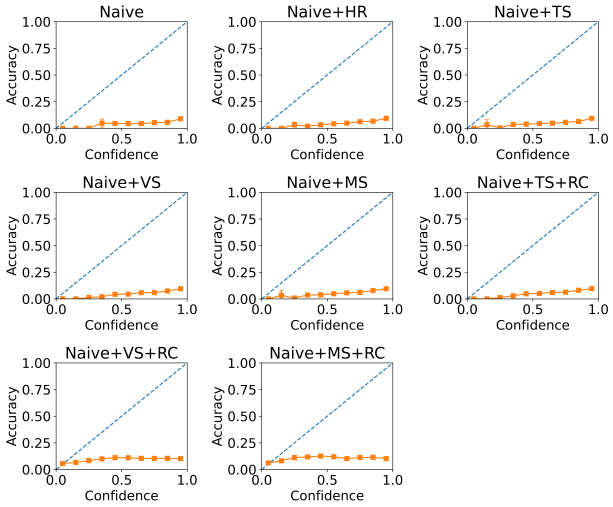

Figure 16. Reliability diagrams for Naive on Split CIFAR100

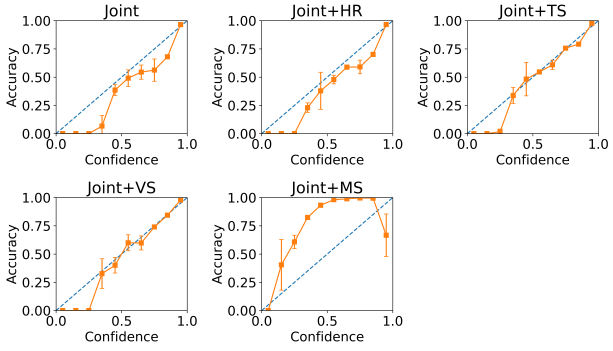

Figure 17. Reliability diagrams for Joint on EuroSAT

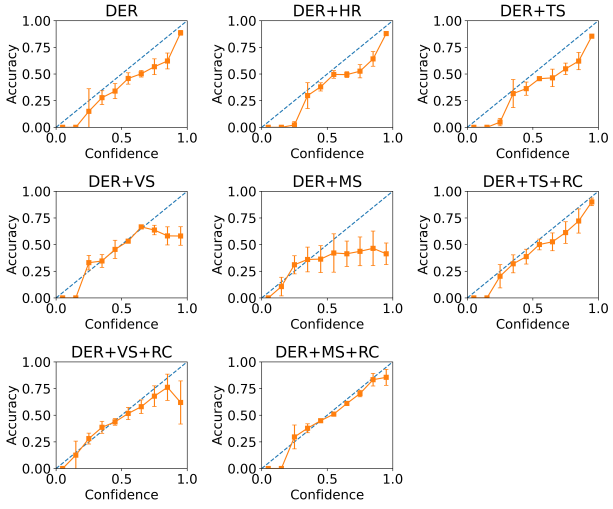

Figure 18. Reliability diagrams for DER++ on EuroSAT

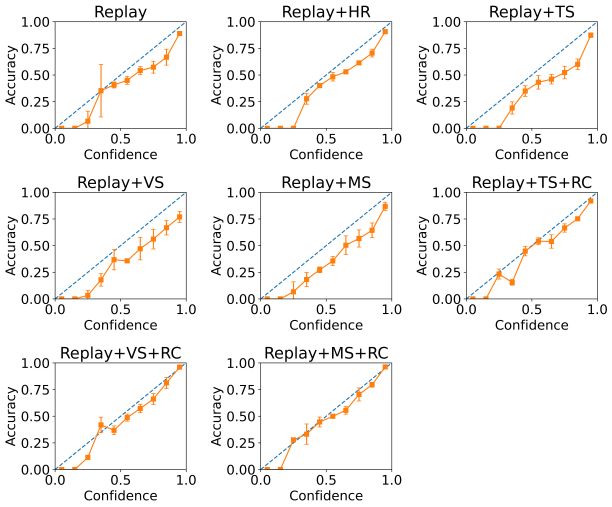

Figure 19. Reliability diagrams for Replay on EuroSAT

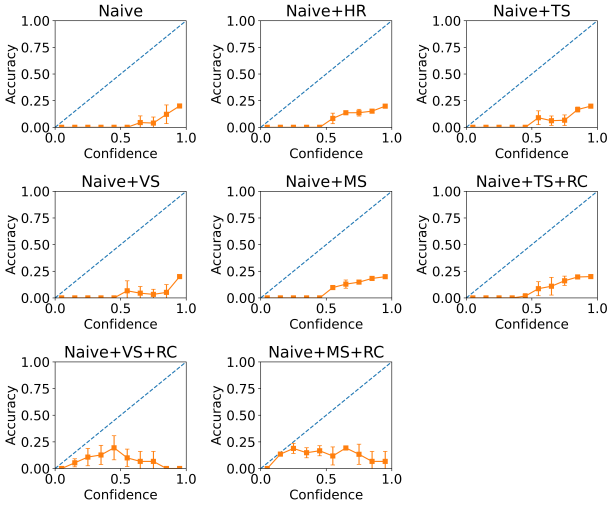

Figure 20. Reliability diagrams for Naive on EuroSAT

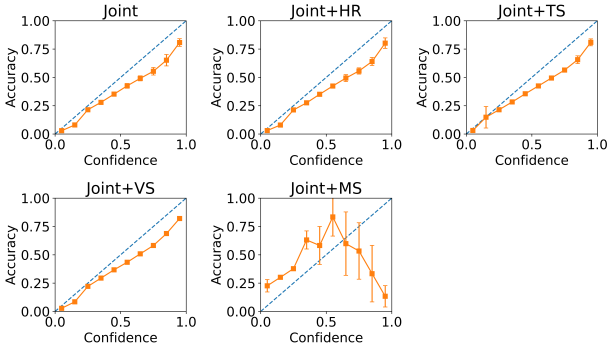

Figure 21. Reliability diagrams for Joint on Atari

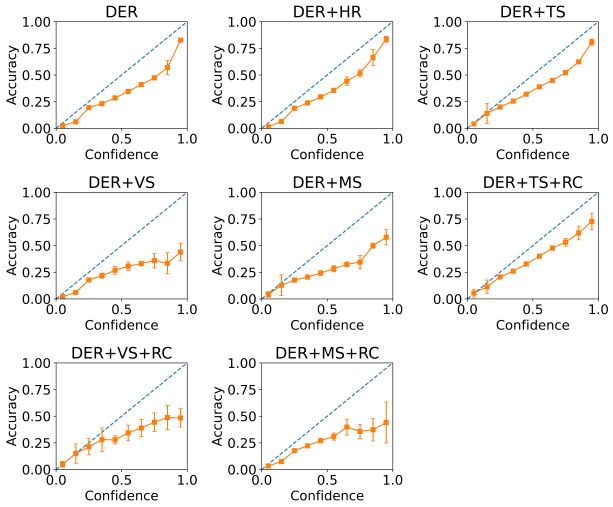

Figure 22. Reliability diagrams for DER++ on Atari

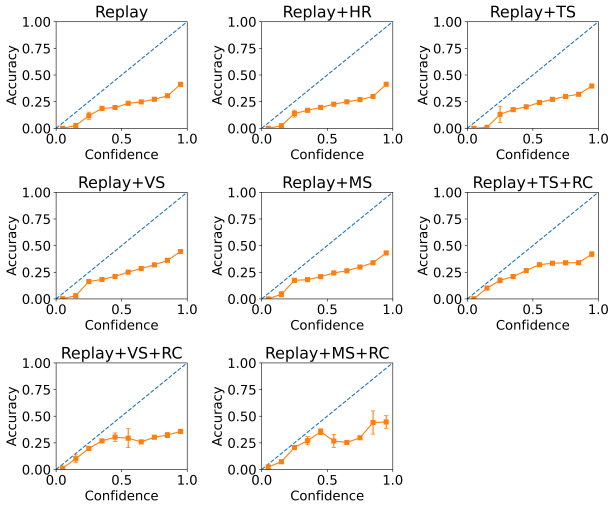

Figure 23. Reliability diagrams for Replay on Atari

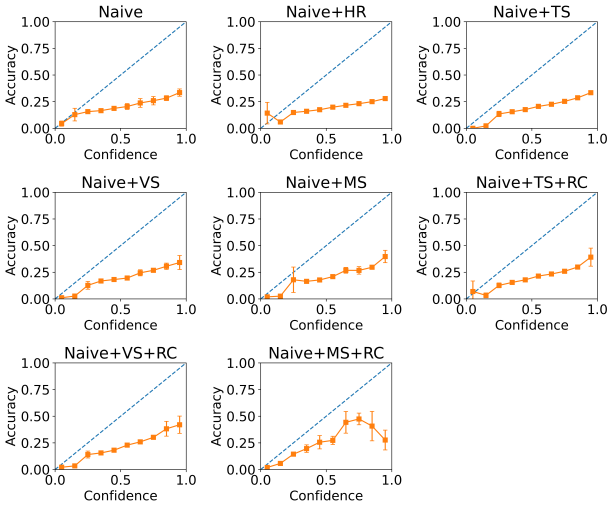

Figure 24. Reliability diagrams for Naive on Atari