Countering Relearning with Perception Revising Unlearning (Supplementary)

Chenhao Zhang CHENHAO.ZHANG@UQ.EDU.AU
University of Queensland
Weitong Chen T.CHEN@ADELAIDE.EDU.AU
Wei Zhang WEI.E.ZHANG@ADELAIDE.EDU.AU
University of Adelaide
Miao Xu[*] MIAO.XU@UQ.EDU.AU
University of Queensland

Editors: Vu Nguyen and Hsuan-Tien Lin

Appendix A. PRU parameter settings

Table 1 lists the training parameters of the PRU in order to accommodate the use of $\mathcal{D}_f$. Table 2 lists the training parameters of the PRU when it is adapted to better fit $\mathcal{D}_f^{new}$.

Table 1: PRU unlearning parameters, when using $\mathcal{D}_f$

	Epoch	lr for E	lr for C
MNIST	10	5×10^{-3}	1×10^{-1}
CIFAR10-AllCNN	10	1×10^{-2}	1×10^{-5}
CIFAR10-ResNet18	10	4×10^{-4}	1×10^{-4}

Table 2: PRU unlearning parameters, when using $\mathcal{D}_f^{new}$

	Epoch	lr for E	lr for C
MNIST	10	1×10^{-2}	1×10^{-1}
CIFAR10-AllCNN	10	7×10^{-2}	1×10^{-1}
CIFAR10-ResNet18	10	1×10^{-4}	1×10^{-2}

The amount of data in $\mathcal{D}_f^{new}$ is significantly smaller than in $\mathcal{D}_f$, which results in a reduced performance in the utility and relearning test for each method. We attempted to tune the training parameters for each method in order to enhance their adaptability to the limited data setting, but the experimental outcomes remained largely unchanged following the adjustments to all the comparison methods. Conversely, PRU is more flexible in terms of making adjustments to the training data, which resulted in a significant improvement in performance on $\mathcal{D}_f^{new}$ after increasing the individual learning rates of PRU. This also demonstrates the flexibility of PRU in comparison to the other methods. It is also noteworthy that, as mentioned in the main text, even without the adjustment of training parameters, PRU outperforms the other methods.

[*] Corresponding author

Table 3: Results when using $\mathcal{D}_f^{new}$. Results in the table show the 0/1/5/10-th relearning epoch result.

Training parameter	Acc	CIFAR-10_AllCNN	CIFAR-10_ResNet
Table 1	$A_r \uparrow$	84.6 / 85.4 / 89.3 / 89.8	84.4 / 90.0 / 90.6 / 90.7
	$A_f \downarrow$	0.9 / 46.2 / 51.8 / 48.0	12.1 / 63.0 / 41.9 / 30.6
Table 2	$A_r \uparrow$	78.0 / 81.7 / 88.3 / 89.2	84.2 / 89.6 / 90.5 / 90.6
	$A_f \downarrow$	0.0 / 0.1 / 0.8 / 1.2	0.7 / 33.4 / 13.5 / 7.4

Appendix B. Case study settings

When testing the PRU on ViT, we fine-tune a pre-trained ViT (ViT_B_16_Weights.IMAGENET1K_V1) on the CIFAR-10 dataset with a batch size of 32, a learning rate of 0.001 for only one epoch. The optimizer is SGD with a momentum of 0.9. Table 4 lists the PRU parameters when using the ViT architecture.

Table 4: PRU unlearning parameters, when using $\mathcal{D}_f$

	Epoch	lr for E	lr for C
CIFAR10-ViT	1	1×10^{-3}	1×10^{-2}

Appendix C. Experiments on more datasets

We include more datasets in this subsection to show the proposed PRU's generalization. Specifically, we further include Fashion-MNIST (Xiao et al., 2017), Kuzushiji-MNIST (Clanuwat et al., 2018), and CIFAR-100 (Krizhevsky et al., 2009) to evaluate the PRU. For datasets except cifar100, we conducted experiments using all ten classes separately as forgetting classes. For the CIFAR-100, there are hundreds of classes. Thus, for the convenience of the experiment, we randomly selected 10 classes in the CIFAR-100 for experiments. The randomly selected CIFAR-100 classes are [45,18,13,33,12,59,58,79,41,5]. We only use AllCNN as the model structure for experiments in this subsection.

Table 5: Classification accuracy of the unlearned model on extra datasets.

Data	Acc	Original	Retrain	Unroll	*UnrollOF*	*BoundShrink*	*BoundExpand*	*PRU (OURS)*
FM	$A_r \uparrow$	89.72	89.56	84.91	77.82	79.33	79.15	**80.16**
	$A_f \downarrow$	89.72	0.00	0.65	6.94	0.37	9.69	**0.00**
KM	$A_r \uparrow$	93.17	93.50	91.41	86.97	90.08	90.05	89.02
	$A_f \downarrow$	93.17	0.00	0.50	1.56	12.81	12.56	**0.00**
C100	$A_r \uparrow$	64.28	65.65	58.46	64.08	56.40	56.38	54.70
	$A_f \downarrow$	64.28	0.00	1.04	55.88	13.24	13.22	**0.02**

As shown in Table 5, the Retrain and Unroll methods utilize retaining data, giving them an advantage over the proposed PRU, which does not use retaining data. Despite this, PRU demonstrates competitive performance in terms of the unlearned model's accuracy

Table 6: Relearning results on extra datasets. Unlearned models are updated with $\mathcal{D}_r^{new}$ in 10 epochs, and the results of the 1/5/10-th epochs are reported.

Data	Acc	Original	Retrain	Unroll	*UnrollOF*	*BoundShrink*	*BoundExpand*	*PRU (OURS)*
FM	$A_r \uparrow$	91.2/92.1/92.3	91.5/92.0/92.1	91.7/92.1/92.3	91.2/92.1/92.3	89.8/92.0/92.3	90.8/92.0/92.2	**89.1/91.8/92.1**
	$A_f \downarrow$	86.5/64.1/48.8	0.0/0.0/0.0	12.1/10.2/6.9	86.7/64.3/48.1	71.3/57.4/44.9	84.9/59.4/43.8	**3.7/5.2/3.4**
KM	$A_r \uparrow$	95.1/96.3/96.7	95.6/96.4/96.7	95.8/96.5/96.8	95.1/96.3/96.7	95.1/96.3/96.7	95.1/96.3/96.7	**93.2/96.1/96.6**
	$A_f \downarrow$	95.7/90.7/82.1	0.0/0.0/0.0	11.7/8.1/4.2	95.6/90.9/82.2	95.7/90.8/82.2	95.7/90.7/82.3	**2.6/4.0/1.5**
C100	$A_r \uparrow$	65.0/65.4/65.6	65.9/66.2/66.3	64.3/65.3/65.5	65.0/65.4/65.5	65.0/65.4/65.5	65.0/65.4/65.5	**59.1/64.3/65.2**
	$A_f \downarrow$	58.9/38.1/22.8	0.0/0.0/0.0	3.3/2.5/1.4	58.9/38.4/22.8	57.9/38.1/22.7	57.7/38.0/22.5	**3.9/5.0/3.8**

performance. In Table 6, a model that relearns slowly should exhibit a slow increase in A_f, which is effectively achieved by PRU and comparable to the gold standard Retrain.

Appendix D. Unlearning time costs

We record the time taken for each method from the beginning of their unlearning to getting the unlearned model under the same running environment. Noteworthy, in Firgure 1, the retraining takes significantly more time than the other methods, so we set the y-axis in a log scale to make it easier to display. Although Unrolling and Boundary Expanding take less time than our method, as mentioned above, our method does not need remaining class data compared with the Unrolling and can solve the relearning problem compared with the Boundary Expanding. In addition to retraining, our method is also faster than Boundary Shrink because it needs to compute the gradient to find the nearest remaining class for the forgetting class sample, which increases the computational complexity.

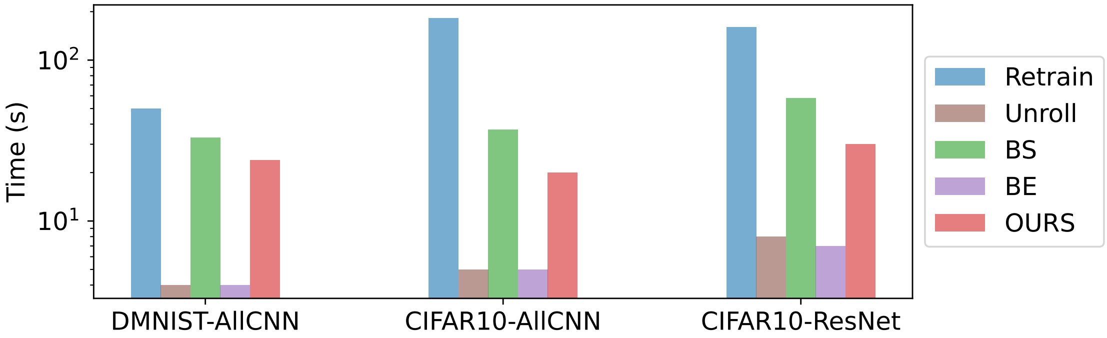

Figure 1: Unlearning time costs in second.

Acknowledgments

The work is supported by the Australian Research Council DE230101116 and DP240103070.

References

Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature, 2018.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.