Overcoming Catastrophic Forgetting for Continual Learning via Feature Propagation

Xuejun Han[1]
xuejunhan@cmail.carleton.ca
Yuhong Guo[1,2]
yuhong.guo@carleton.ca

[1] Carleton University
Ottawa, Canada

[2] Canada CIFAR AI Chair, Amii
Edmonton, Canada

1 Dataset Statistics

Table 1 summarizes the statistics of all benchmarks.

Dataset	# tasks	# classes per task	# images per task	input size
S-MNIST	5	2	12000	$1 \times 28 \times 28$
P-MNIST	20	10	1000	$1 \times 28 \times 28$
R-MNIST	20	10	1000	$1 \times 28 \times 28$
S-CIFAR-10	5	2	10000	$3 \times 32 \times 32$
S-CIFAR-100	20	5	2500	$3 \times 32 \times 32$
S-Tiny-ImageNet	10	20	10000	$3 \times 64 \times 64$

Table 1: The Statistics of datasets.

2 Metrics

For a principled evaluation, we consider two metrics, *average accuracy* (↑) and *forgetting measure* (↓) [1], to evaluate the performance of models on test data of all tasks. The average accuracy evaluates the overall performance for all seen tasks, while the forgetting measure reflects the accuracy drops on previous tasks after the model is trained on new tasks. The large forgetting values signify the model has less stability when encountering new tasks.

Suppose we have a sequence of tasks with identifier $t \in \{1, \cdots, T\}$ and denote $R_{i,j}$ the classification accuracy of the model on test data of task j after learning the training data up till task i. The two metrics are defined as follows,

2.1 Average Accuracy (↑)

$$ACC_t = \frac{1}{t} \sum_{k=1}^{t} R_{t,k} \tag{1}$$

The above definition indicates the average accuracy on test data of task 1 to t after the model has learned continually up till task t, the values of which are used to draw the accuracy

	S-MNIST		**S-CIFAR-10**		**S-CIFAR-100**		**R-MNIST**	**P-MNIST**
Model	Class-IL	Task-IL	Class-IL	Task-IL	Class-IL	Task-IL	Domain-IL	Domain-IL
Finetune	99.29	4.81	80.09	21.32	62.84	26.13	19.63	22.25
ER-Res.	28.21	0.68	49.37	3.15	60.45	10.83	8.77	15.74
GEM	21.38	1.24	80.17	6.88	58.74	14.33	8.82	18.57
A-GEM	66.72	0.95	82.76	13.49	62.61	17.49	15.61	21.27
GSS	34.89	1.12	59.12	7.13	61.72	8.00	10.84	23.44
FDG	24.10	0.82	78.61	12.19	64.39	23.22	8.04	11.61
HAL	24.77	0.61	39.29	6.58	43.11	11.27	8.97	**11.25**
SCR	–	–	**31.23**	5.91	43.27	8.17	–	–
CCL-FP	**8.43**	0.22	34.11	2.63	41.86	8.92	7.69	15.12
CCL-FP+	8.45	**0.18**	34.41	**2.05**	**41.61**	**7.95**	**6.48**	14.88

Table 2: The average forgetting (%) for baselines and our models across 5 runs with different random seeds on selected datasets. The best results on each benchmark are marked in bold.

	S-MNIST		**S-CIFAR-10**		**S-CIFAR-100**		**S-Tiny-ImageNet**		**P-MNIST**	**R-MNIST**
Setup	Class-IL	Task-IL	Class-IL	Task-IL	Class-IL	Task-IL	Class-IL	Task-IL	Domain-IL	Domain-IL
Joint-online	95.59	99.33	58.89	87.58	19.60	69.80	14.21	43.89	77.65	84.12
Joint-offline	97.64	99.73	91.58	98.17	70.87	95.42	57.40	81.28	74.36	90.54

Table 3: The average accuracy across five runs with different random seeds of Joint training in the single epoch and multiple epochs settings.

evolution curve shown in Figure 2 (paper). In particular, ACC_T is the average accuracy on test data of all tasks after completing the whole continual training, of which the results on all benchmarks are given in Table 1 (paper).

2.2 Forgetting Measure (↓)

$$FGT_t = \frac{1}{t-1}\sum_{k=1}^{t-1} r_{t,k} \tag{2}$$

where $r_{t,k}$ is the forgetting on task k after the model has be trained continually up till task t, which is calculated as follows,

$$r_{t,k} = \max_{i\in\{1,\cdots,t-1\}} R_{i,k} - R_{t,k} \tag{3}$$

The forgetting measures the accuracy drops on previous tasks after the model is trained on new tasks. The large forgetting values signify the model has less stability when encountering new tasks. The results for forgetting measure are reported in Table 2. CCL-FP+ produces the best performance on a majority of cases compared to other CL competitors, demonstrating the overall outstanding ability of our model to overcome catastrophic forgetting even with a very small buffer.

3 Upper Bound

Single epoch training is a very compelling and ideal setup for continual learning and somewhat close to the spirit why we focus on general continual learning [6]. However, Joint, as an upper bound for continual learning, suffers underfitting to some extent in this setting. In Table 3, we compare the results of joint training in the online (single-epoch) and offline (multi-epoch) setups.

	M=0.2k		M=0.5k		M=1k		M=5k	
$\lvert\mathcal{B}_{\mathcal{M}}\rvert$	SCR	Ours	SCR	Ours	SCR	Ours	SCR	Ours
10	40.91	50.11	45.19	57.03	46.05	60.29	46.12	63.84
20	38.85	48.24	48.83	56.89	49.24	63.63	52.67	67.17
50	31.26	43.77	45.66	54.43	57.27	65.77	64.72	72.33
100	30.31	44.08	45.29	53.18	55.78	62.37	68.67	75.86
200	31.28	37.57	43.68	49.25	55.43	58.47	72.91	75.17

Table 4: The average accuracy (%) for our model CCL-FP+ and SCR across 5 runs with different random seeds. Part of numbers in this table are used to generate Figure 4 (paper) and Table 4 (paper).

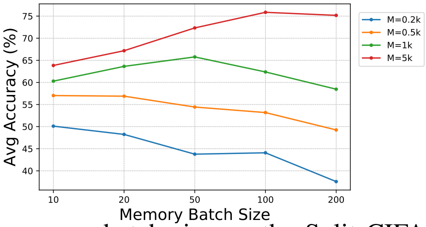

Figure 1: Impact of the memory batch size on the Split CIFAR-10 dataset in the class incremental setting. M is the memory buffer size.

4 Comparison with SCR

We further study the impact of different memory batch size in cases of different buffer size for our model CCL-FP+ and SCR on the Split CIFAR-10 dataset in the class incremental setting. As we can see in Table 4, the small memory batch size is preferable when the buffer size is small, e.g. the performance achieves the best when $|\mathcal{B}_{\mathcal{M}}| = 10$ for M=0.2k and starts to drop when $|\mathcal{B}_{\mathcal{M}}| \geq 20$. For M=1k, $|\mathcal{B}_{\mathcal{M}}| = 50$ is preferable and for M=5k, $|\mathcal{B}_{\mathcal{M}}| = 100$ is desirable.

Generally speaking, contrastive representation learning benefits from larger memory batch size since it means more negative samples [3, 4, 5]. As we can see in Figure 1, when the buffer size is large, e.g. M=5k, the performance significantly improves with the increase of memory batch size and achieves the best when $\mathcal{B}_{\mathcal{M}} = 100$. However, in the case of small buffer size, e.g. M=0.2k, large memory batch size can observably degrade the performance because the model may easily overfit the memory data. Furthermore, we compare our model CCL-FP+ with SCR [5] on S-CIFAR-10 and S-CIFAR-100 datasets in the class incremental setting with memory batch size of 10 and 100. As shown in Table 5, our model CCL-FP+ consistently outperforms SCR in all cases.

5 Hyperparameter Sensitivity

We report the sensitivity of CCL-FP+ to hyperparameters w, α, β in terms of average accuracy on the Split CIFAR-10 dataset in the class incremental setting in Table 6. As shown, a larger value of w contributes to better performance demonstrating the appreciable impact of feature propagation. The model is not overly sensitive to α and β. Generally, $\alpha \leq 0.5$ yields decent performance. The large value of α may degrade the performance as a result of overly protecting the representation space for past tasks and impairing the strength to adapt to new

	M=0.2k		M=0.5k		M=1k		M=5k	
Model	10	100	10	100	10	100	10	100
SCR	40.91	30.01	45.19	45.29	46.05	55.78	46.12	68.67
Ours	50.11	44.08	57.03	53.18	60.29	62.37	63.84	75.86
SCR	7.38	4.31	9.97	6.83	12.06	12.38	14.71	37.67
Ours	14.05	9.36	19.44	14.47	23.99	20.76	27.38	39.43

Table 5: The comparison of average accuracy of SCR and CCL-FP+ with different memory batch size (10 vs. 100) on Split CIFAR-10 (up) and Split CIFAR-100 (down) datasets in the class incremental setting. M is the memory buffer size.

w	ACC	α	ACC	β	ACC
0.1	46.73	0.1	51.74	0.1	51.08
0.3	50.98	0.5	49.42	0.5	51.74
0.5	51.74	1	48.91	1	50.93

Table 6: Hyperparameter sensitivity of CCL-FP+ to w, α, β in terms of average accuracy in the class incremental setting on the Split CIFAR-10 dataset.

tasks. The model is not sensitive to β as long as it falls into a reasonable range of $[0,1]$.

6 2D t-SNE Visualization

Figure 2 presents the visualization results of CCL-FP+ and ER on P-MNIST. Each class/color is consist of images from 20 tasks. The representations learned by CCL-FP+ are better separable compared with ER.

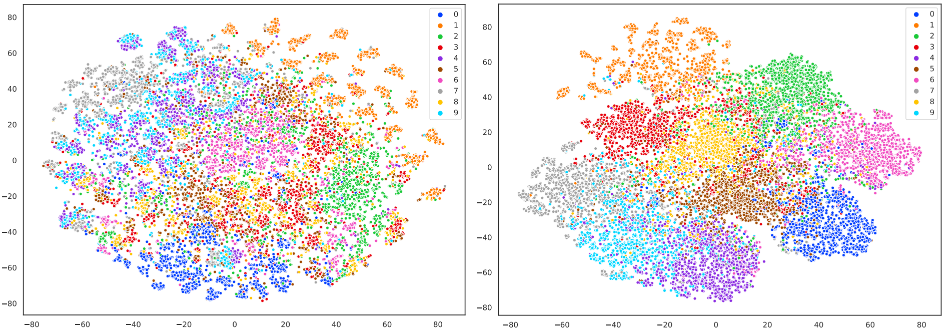

Figure 2: 2D t-SNE visualization of data embeddings of P-MNIST at the end of 20 tasks. **Left**: ER; **Right**: CCL-FP+.

7 Algorithm

Algorithm 1 provides a pseudocode for CCL-FP+. CCL-FP is easily achieved by removing the step of supervised contrastive loss. Algorithm 2 describes the procedure of reservoir sampling [2, 7].

Algorithm 1 Training of CCL-FP+

```
Input: Sequential dataset $\mathcal{D} = \{\mathcal{D}_1, \cdots, \mathcal{D}_T\}$, batch size $b$, learning rate $\lambda$, scalars $w$, $\alpha$, $\beta$
$\mathcal{M} \leftarrow \{\}$    // initialize memory set
$\theta_o \leftarrow$ None, $\theta \leftarrow$ Rand. Init.    // define $\theta_o$ and initialize model parameters $\theta$
$\theta$: parameters of $f$ and $\psi$    // classifier $f$ and feature extractor $\psi$
for $t \in \{1, \cdots, T\}$ do
    for $\mathcal{B} \sim \mathcal{D}_t$ do    // sample a minibatch of size $b$ from current dataset
        $\mathcal{B}_{\mathcal{M}} \sim \mathcal{M}$    // sample a minibatch of size $b$ from memory set
        if $\theta_o$ is None then
            $\mathcal{L}_{ce} \leftarrow l_{ce}(f(\psi(\mathcal{B} \cup \mathcal{B}_{\mathcal{M}})))$    // standard cross-entropy loss
            $\mathcal{L}_{scl} \leftarrow l_{scl}(\psi(\mathcal{B} \cup \mathcal{B}_{\mathcal{M}}))$    // supervised contrastive loss
            $\mathcal{L} \leftarrow \mathcal{L}_{ce} + \beta \cdot \mathcal{L}_{scl}$
        else
            $\tilde{\psi}(\mathcal{B} \cup \mathcal{B}_{\mathcal{M}}) \leftarrow w \cdot \psi(\mathcal{B} \cup \mathcal{B}_{\mathcal{M}}) + (1-w) \cdot \mathrm{A}\psi_o(\mathcal{B} \cup \mathcal{B}_{\mathcal{M}})$    // modify representation using feature propagation
            $\mathcal{L}_{ce} \leftarrow l_{ce}(f(\tilde{\psi}(\mathcal{B} \cup \mathcal{B}_{\mathcal{M}})))$    // the cross-entropy loss on modified representation
            $\mathcal{L}_{cl} \leftarrow l_{cl}(\psi(\mathcal{B} \cup \mathcal{B}_{\mathcal{M}}), \psi_o(\mathcal{B} \cup \mathcal{B}_{\mathcal{M}}))$    // contrastive loss on current and previous representation
            $\mathcal{L}_{scl} \leftarrow l_{scl}(\psi(\mathcal{B} \cup \mathcal{B}_{\mathcal{M}}))$    // supervised contrastive loss
            $\mathcal{L} \leftarrow \mathcal{L}_{ce} + \alpha \cdot \mathcal{L}_{cl} + \beta \cdot \mathcal{L}_{scl}$
        end if
        $\theta \leftarrow \theta - \lambda \cdot \nabla_{\theta}\mathcal{L}$    // single SGD step to update parameters
        $\mathcal{M} \leftarrow reservoir(\mathcal{M}, \mathcal{B})$    // update memory set using reservoir sampling
    end for
    $\theta_o \leftarrow \theta$    // update the parameters of $\theta_o$ to $\theta$
end for
return $\theta$
```

Algorithm 2 Reservoir Sampling

```
Input: The memory set $\mathcal{M}$, number of seen examples $N$, example $(\mathbf{x}, y, t)$
if $|\mathcal{M}| > N$ then
    $\mathcal{M}[N] \leftarrow (\mathbf{x}, y, t)$
else
    $i = randint(0, N)$
    if $i < |\mathcal{M}|$ then
        $\mathcal{M}[i] \leftarrow (\mathbf{x}, y, t)$    // overwrite memory slot
    end if
end if
```

References

[1] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *European Conference on Computer Vision (ECCV)*, 2018.

[2] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. Continual learning with tiny episodic memories. In *International Conference on Machine Learning (ICML)*, 2019.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

[4] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[5] Zheda Mai, Ruiwen Li, Hyunwoo Kim, and Scott Sanner. Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning. In *CVPR workshop*, 2021.

[6] Buzzega Pietro, Boschini Matteo, Porrello Angelo, Abati Davide, and Calderara Simone. Dark experience for general continual learning: a strong, simple baseline. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[7] Jeffrey S. Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software*, 11(1):37–57, 1985.