ACIL: Active Class Incremental Learning for Image Classification: Supplemental File

Aditya R. Bhattacharya[*]
arb17b@fsu.edu
Debanjan Goswami[*]
dgoswami@fsu.edu
Shayok Chakraborty
shayok@cs.fsu.edu

Department of Computer Science
Florida State University
Tallahassee, Florida, USA

Abstract

In this Supplemental File, we include the following:

1. The pseudo-code of the proposed *ACIL* algorithm (Section 1)
2. The experimental setup details (Section 2)
3. Study of the backbone network architecture (Section 3)
4. Performance with varying number of episodes (Section 4)
5. Computation time analysis (Section 5)
6. Ablation studies (Section 6)
7. Performance analysis using the Retention metric (Section 7)

1 Pseudo-code of the ACIL Algorithm

The pseudo-code of the proposed *ACIL* algorithm is depicted in Algorithm 1. As evident from the pseudo-code, our algorithm is computationally lightweight, simple and easy to implement.

2 Experimental Setup Details

Table 1 details the following numbers for each dataset used in our study: (i) number of labeled samples (X_n^L) in each episode; (ii) number of unlabeled samples (X_n^U) in each episode; (iii) budget for the exemplar set (E_n) in each episode. The average values over 3 random trials are reported.

[*]These authors contributed equally to this work

Algorithm 1 The Proposed *ACIL* Framework

```
Require: A DNN architecture for image classification, exemplar set budget $k$ for each episode, weight parameter $\lambda$

1: for $n = 1, 2, \ldots N$ do
2:     Receive the labeled data $X_n^L$ and the unlabeled data $X_n^U$ in episode $n$. Let $C_{episode}$ denote the set of classes in $X_n^L$
3:     Receive the annotated exemplar set $E_{n-1}$ from episode $n-1$. Let $C_{exemplar}$ denote the set of classes in $E_{n-1}$
4:     Train a DNN on $X_n^L$ and $E_{n-1}$ using the loss function in Equation (8) of the main paper
5:     Apply the trained network on $X_n^U$ to derive the pseudo-labels of these samples.
6:     Split the budget $k$ into $k_{unlabeled}$ and $k_{exemplar}$, as detailed in Section 3.2 of the main paper
7:     Select $\frac{k_{unlabeled}}{|C_{episode}|}$ samples from each class in $X_n^U$ (given by the pseudo-labels) using the active sampling strategy (Section 3.3 of the main paper)
8:     Select $\frac{k_{exemplar}}{|C_{exemplar}|}$ samples from each class in $E_{n-1}$ using the active sampling strategy (Section 3.3 of the main paper)
9:     Merge all the selected samples to derive the exemplar set $E_n$ selected in episode $n$
10:    Append the exemplar set $E_n$ to the data in the next episode $(n+1)$
11: end for
```

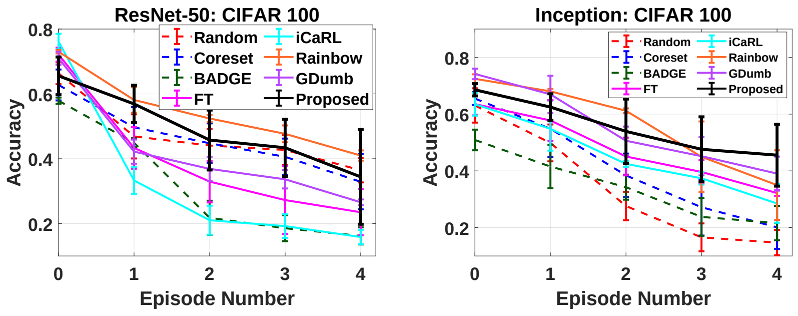
	X_n^L	X_n^U	E_n
MNIST	$2,400 \pm 92$	$9,600 \pm 364$	$1,200$
CIFAR 10	$2,000 \pm 0$	$8,000 \pm 0$	$1,000$
SVHN	$2,930 \pm 1,071$	$11,721 \pm 5,354$	$2,000$
COIL	114 ± 0	456 ± 0	100
CIFAR 100	$1,000 \pm 0$	$4,000 \pm 0$	500
Tiny ImageNet	200 ± 0	800 ± 0	500

Table 1: Details of the experimental setup for each dataset used in our study. The average values over 3 random trials are reported in the table.

3 Study of the backbone network architecture

In this experiment, we studied the effect of the backbone deep neural network architecture on the performance of *ACIL*. The results on the CIFAR 100 dataset using the ResNet-50 [2] and Inception [3] architectures are depicted in Figure 1. The number of samples that had to be manually annotated per episode are depicted in Table 2 (these values are the same as those in Table 1 in the main paper, since only the backbone architecture is changed; they are included here for ease of analysis). The results follow a similar trend, with *ACIL* showing very similar accuracy curves as the CIL baselines, and better than the AL baselines. For the Inception backbone, *ACIL* depicts the highest accuracy at the end of the last episode. The AL baselines *Random* and *Coreset* depict impressive performance for the ResNet-50 backbone, but their performance is much worse for the Inception backbone.

From Table 2, we once again note that our framework can result in tremendous reduction in the human annotation effort, compared to the CIL baselines. It also necessitates fewer annotations than the AL baselines, and produces much better accuracy results than them. These results corroborate the robustness of *ACIL* to the backbone deep network architecture.



(a) ResNet-50 Backbone

(b) Inception Backbone

Figure 1: Study of the backbone network architecture on the CIFAR 100 dataset. The AL baselines (*Random, Coreset, BADGE*) are shown with dotted lines; the CIL baselines (*Fine-tuning, iCaRL, Rainbow, GDumb*) and the proposed *ACIL* method are shown with solid lines. Best viewed in color.

	CIL Baselines	**AL Baselines**	**Proposed**
CIFAR 100	$5{,}000 \pm 0.00$	$1{,}450 \pm 152.56$	$1{,}158 \pm 131.53$

Table 2: Average ($\pm$ std) number of samples that needed to be annotated per episode (including the episodic labeled set X_n^L) by the CIL baselines (*Finetuning, iCaRL, Rainbow, GDumb*), AL baselines (*Random, Coreset, BADGE*) and the proposed method (*ACIL*). These values are the same as those in Table 1 in the main paper, since only the backbone architecture is changed; they are included here for ease of analysis.

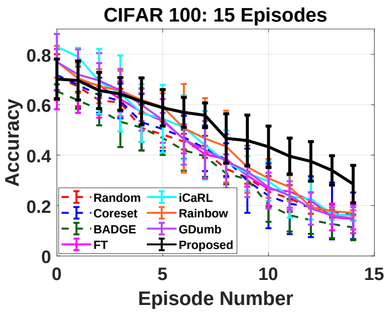
4 Performance with varying number of episodes

The goal of this experiment was to study the performance of *ACIL* with varying number of total episodes (and hence, varying number of classes per episode). The results on the CIFAR 100 dataset with 15 and 25 episodes are depicted in Figure 2 (the results with 5 episodes are presented in Figure 2(e) in the main paper). From Figure 2, we note that the proposed *ACIL* method depicts comparable performance to the class incremental learning baselines (*Finetuning, iCaRL, Rainbow, GDumb*) and outperforms the active learning baselines (*Random, Coreset, BADGE*). In most of the episodes across the two experiments, the accuracy furnished by *ACIL* is very close to (or better than) that of the CIL baselines, and much better than the AL baselines. This is consistent with the results in the main paper. For both these experiments, *ACIL* attains the highest accuracy after the last episode.

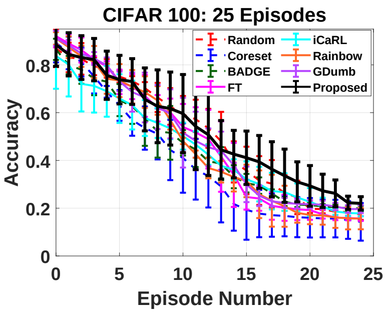
As before, from Table 2, we note that *ACIL* results in substantial savings of the human annotation effort compared to the CIL baselines; it also requires fewer annotations than the AL baselines. These results show the robustness of *ACIL* to the varying number of episodes and the varying number of classes per episode.

5 Computation Time Analysis

In this section, we present our analysis of the computation time of all the methods studied. The results are presented in Table 3 (for the AL baselines *Random, Coreset, BADGE*) and Table 4 (for the CIL baselines (*Finetuning, iCaRL, Rainbow, GDumb*) and the proposed



(a) CIFAR 100: 15 Episodes



(b) CIFAR 100: 25 Episodes

Figure 2: Study of varying number of total episodes on the CIFAR 100 dataset. The AL baselines (*Random, Coreset, BADGE*) are shown with dotted lines; the CIL baselines (*Fine-tuning, iCaRL, Rainbow, GDumb*) and the proposed *ACIL* method are shown with solid lines. Best viewed in color.

method (*ACIL*)). The reported time includes the time taken for data selection **and** training the deep neural network in a given episode (averaged over all the episodes and over 3 random trials for each dataset). The algorithms were implemented in Python on a Linux workstation with Intel(R) Xeon(R) Gold 5222 CPU @ 3.80GHz and 64GB RAM, equipped with Dual NVIDIA Quadro RTX 5000 GPUs with 16GB memory.

	Random	**Coreset**	**BADGE**
MNIST	17.75 ± 5.49	31.86 ± 2.98	24.58 ± 2.22
CIFAR 10	23.33 ± 6.28	36.79 ± 6.09	29.44 ± 4.06
SVHN	42.59 ± 18.09	56.13 ± 16.43	62.13 ± 12.16
COIL	29.14 ± 9.62	41.79 ± 7.66	38.69 ± 6.51
CIFAR 100	22.61 ± 10.67	34.70 ± 7.16	31.33 ± 8.52
Tiny ImageNet	18.17 ± 4.34	26.84 ± 4.66	35.34 ± 8.48

Table 3: Average computation time (in seconds) per episode for the AL baselines (*Random, Coreset, BADGE*). The results are averaged over all the episodes and over 3 random trials for each dataset.

	FT	**iCaRL**	**Rainbow**	**GDumb**	**Proposed**
MNIST	50.65 ± 7.58	62.25 ± 4.94	72.71 ± 29.46	64.29 ± 21.01	58.06 ± 14.60
CIFAR 10	83.11 ± 22.70	106.34 ± 9.13	63.75 ± 7.11	64.30 ± 8.01	42.90 ± 10.93
SVHN	108.90 ± 46.71	132.70 ± 13.06	101.74 ± 44.62	100.63 ± 26.09	71.09 ± 30.93
COIL	47.11 ± 17.27	54.23 ± 8.55	77.79 ± 27.25	73.06 ± 23.43	65.68 ± 18.61
CIFAR 100	58.62 ± 17.83	47.04 ± 9.92	54.28 ± 11.22	50.99 ± 12.11	36.98 ± 10.71
Tiny ImageNet	40.42 ± 7.81	44.62 ± 8.09	41.06 ± 6.14	41.94 ± 8.76	32.82 ± 6.91

Table 4: Average computation time (in seconds) per episode for the CIL baselines (*Finetuning, iCaRL, Rainbow, GDumb*) and the proposed method (*ACIL*). The results are averaged over all the episodes and over 3 random trials for each dataset.

We note that the CIL baselines, in general, have a marginally higher computation time than the other methods. This is because, in each episode, the deep neural network has to be trained on *all the samples* in that episode (in addition to the exemplar set from the previous episode), which increases the computation time. For the proposed method, the deep model has to be trained only on the labeled set X_n^L and the exemplar set E_{n-1} in a given episode n; it therefore has lower computational overhead than the CIL methods. The same applies for the AL baselines, which also have low computational overhead. The proposed method depicts slightly higher computation time than the AL baselines and comparable (and sometimes slightly lower) computation time as the CIL baselines. This further shows the usefulness of *ACIL* for real-world applications.

6 Ablation Studies

We conducted ablation studies to analyze the importance of the uncertainty and diversity components in our *ACIL* framework. The results on the SVHN dataset are depicted in Figure 3. We note that the performance of *ACIL* is affected when either the uncertainty or the diversity component is excluded from the framework. Excluding the diversity component may result in querying informative, but redundant samples in the exemplar set; excluding the uncertainty component may fail to capture the most informative samples in the exemplar set. These results show the importance of both the components in our framework.

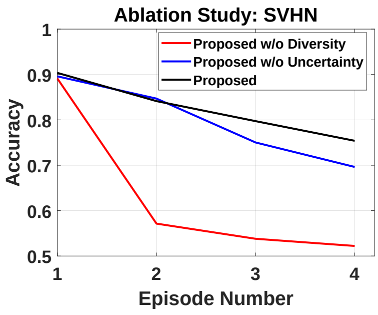


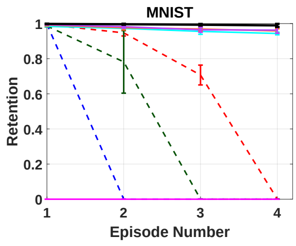
Figure 3: Ablation studies on the SVHN dataset. Best viewed in color.

7 Performance Analysis using the Retention Metric

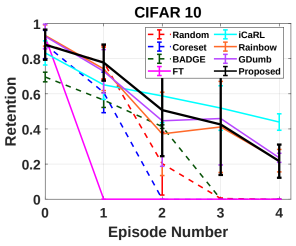
Several metrics have been proposed to quantify catastrophic forgetting in the incremental learning literature [4, 5]. In this section, we demonstrate the efficacy of our framework to mitigate catastrophic forgetting using the *retention* metric. This metric quantifies how much information the model retains from the first episode; after training the deep model in episode n, the retention is computed as the accuracy of the model on the test set from the first episode. The results are presented in Figure 4 and Table 5 (this table as the same as Table 1 in the main paper, and is included here for ease of analysis).

	CIL Baselines	**AL Baselines**	**Proposed**
MNIST	$12,000 \pm 478.12$	$3,359.6 \pm 483.43$	$2,899.6 \pm 447.38$
CIFAR 10	$10,000 \pm 0.00$	$2,800 \pm 414.04$	$2,417.2 \pm 344.90$
SVHN	$14,651.40 \pm 5,542.22$	$4,530 \pm 1,181.98$	$3,763.60 \pm 1,455.34$
COIL	570 ± 0.00	204 ± 30.51	153.33 ± 25.38
CIFAR 100	$5,000 \pm 0.00$	$1,450 \pm 152.56$	$1,158 \pm 131.53$
Tiny ImageNet	$1,000 \pm 0.00$	299 ± 9.97	232.07 ± 21.92

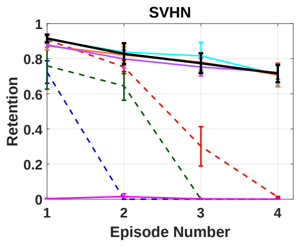
Table 5: Average number of samples that needed to be annotated per episode (including the episodic labeled set X_n^L) by the CIL baselines (*Finetuning, iCaRL, Rainbow, GDumb*), AL baselines (*Random, Coreset, BADGE*) and the proposed method (*ACIL*). This table as the same as Table 1 in the main paper, and is included here for ease of analysis.



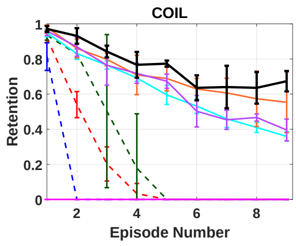
(a) MNIST



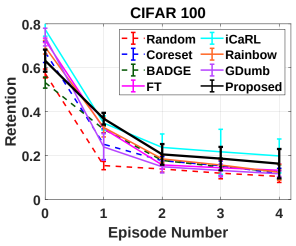
(b) CIFAR 10



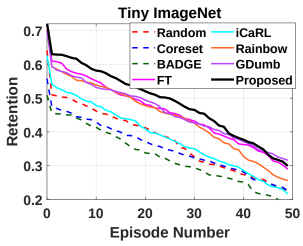
(c) SVHN



(d) COIL



(e) CIFAR 100



(f) Tiny ImageNet

Figure 4: Performance analysis of *ACIL* using the Retention metric. The AL baselines (*Random, Coreset, BADGE*) are shown with dotted lines; the CIL baselines (*Finetuning, iCaRL, Rainbow, GDumb*) and the proposed *ACIL* method are shown with solid lines. The error bars have been omitted from the Tiny ImageNet results for better visualization. Best viewed in color.

Retention: *ACIL* vs. AL baselines: From Figure 4, we note that the proposed method consistently outperforms the AL baselines in terms of retention. The AL baselines select an exemplar set in each episode completely from the unlabeled set of the corresponding episode, as proposed in [■]; thus, they fail to capture the knowledge from the former episodes and hence are not effective in mitigating catastrophic forgetting.

Retention: *ACIL* vs. CIL baselines: *ACIL* once again depicts comparable performance as the CIL baselines. Our framework selects an exemplar set in each episode, based on an uncertainty and diversity based criterion, and is thus able to retain useful information about the former episodes, which enables it to mitigate catastrophic forgetting. Thus, the retention drops at more or less the same rate as the CIL baselines, with increasing number of episodes.

Annotation effort: As noted in Table 5 (and in the main paper), *ACIL* results in substantial savings of annotation effort compared to the CIL baselines, as they require all the samples to be annotated in each episode; *ACIL* is also marginally better than the AL baselines in terms of the annotation effort. These observations further corroborate the promise and potential of *ACIL* for real-world class incremental learning applications.

8 Acknowledgment

This research was supported in part by the National Science Foundation under Grant Number: IIS-2143424 (NSF CAREER Award).

References

[1] E. Belouadah, A. Popescu, U. Aggarwal, and L. Saci. Active class incremental learning for imbalanced datasets. In *European Conference on Computer Vision Workshop (ECCV-W)*, 2020.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[4] Z. Wang, E. Yang, L. Shen, and H. Huang. A comprehensive survey of forgetting in deep learning beyond continual learning. In *arXiv:2307.09218v2*, 2023.

[5] D. Zhou, Q. Wang, Z. Qi, H. Ye, D. Zhan, and Z. Liu. Deep class-incremental learning: A survey. In *arXiv:2302.03648v1*, 2023.