ACIL: Active Class Incremental Learning for Image Classification

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

Continual learning (or class incremental learning) is a realistic learning scenario for computer vision systems, where deep neural networks are trained on episodic data, and the data from previous episodes are generally inaccessible to the model. Existing research in this domain has primarily focused on avoiding catastrophic forgetting, which occurs due to the continuously changing class distributions in each episode and the inaccessibility of the data from previous episodes. However, these methods assume that all the training samples in every episode are annotated; this not only incurs a huge annotation cost, but also results in a wastage of annotation effort, since most of the samples in a given episode will not be accessible to the model in subsequent episodes. Active learning algorithms identify the salient and informative samples from large amounts of unlabeled data and are instrumental in reducing the human annotation effort in inducing a deep neural network. In this paper, we propose ACIL, a novel active learning framework for class incremental learning settings. We exploit a criterion based on uncertainty and diversity to identify the exemplar samples that need to be annotated in each episode, and will be appended to the data in the next episode. Such a framework can drastically reduce annotation cost and can also avoid catastrophic forgetting. Our extensive empirical analyses on several vision datasets corroborate the promise and potential of our framework against relevant baselines.

1 Introduction

Deep neural networks (DNNs) have pushed the boundaries of computer vision and have achieved state-of-the-art performance in a variety of applications [11, 19, 40]. In many real-world applications, the training data arrives in episodes over time, with each episode containing samples from a different set of classes. Further, this episodic data is ephemeral and cannot be held for long due to storage and privacy constraints [12, 20, 22]. The DNN is expected to learn incrementally about the new classes, from their episodic data; at the same time, it should retain knowledge about the formerly learned classes in previous episodes, whose data is no longer accessible. This learning paradigm is called *Class Incremental*

*These authors contributed equally to this work

Learning (CIL), which aims to continually induce a holistic classifier among all the classes encountered. A problem is CIL is *catastrophic forgetting*, which occurs when optimizing the network to recognize the new classes erases its knowledge about the former classes (in previous episodes), resulting in a degradation in performance. Thus, developing algorithms to mitigate catastrophic forgetting has been the primary focus in CIL research [47].

However, another aspect that has been overlooked in this context is the cost of data annotation; CIL algorithms assume that all the training samples in each episode are annotated with their class labels. This incurs a huge cost as data annotation is an expensive process in terms of time, labor and human expertise. Moreover, this results in a wastage of annotation effort, since most of the data samples in a given episode will not be accessible to the deep model in subsequent episodes. This necessitates a strategy to more efficiently utilize the human annotation effort in CIL settings.

Active Learning (AL) algorithms have gained popularity in reducing the human annotation effort in inducing a machine learning / deep learning model [38]. When faced with large amounts of unlabeled data, these algorithms automatically identify the salient and the most informative samples that need to be annotated manually. AL has been successfully used in a variety of applications, such as computer vision [45], text mining [42], medical diagnosis [44] and bioinformatics [30] among others.

In this paper, we introduce *ACIL*, a novel **A**ctive learning framework for **C**lass **I**ncremental **L**earning. Contrary to existing CIL settings, we assume that in each episode, only a small portion of the data is annotated; majority of the samples are unlabeled in each episode. In each episode, we are allowed to select an exemplar set of size k (where k is a pre-determined number) which will be manually annotated and appended to the data in the next episode. The exemplar set can contain samples from the labeled and unlabeled sets in the current episode as well as samples from the exemplar set obtained from the previous episode. We formulate a criterion based on diversity and uncertainty and use weighted set partitioning techniques to select the set of exemplar samples. Such a framework can drastically reduce the data annotation cost, as only a small fraction of the samples is annotated in each episode. It can also potentially avoid catastrophic forgetting, as the exemplar samples in each episode are propagated to subsequent episodes.

2 Related Work

Class Incremental Learning: Class Incremental Learning (CIL) has attracted significant research attention in recent years in the computer vision and machine learning communities; please refer to [47] for a detailed taxonomy and survey. CIL can be broadly organized into three categories: *data-centric, model-centric* and *algorithm-centric*. Data-centric CIL methods attempt to alleviate catastrophic forgetting with the help of extra data, called exemplars, and are further categorized into data replay and data regularization. Replay based methods attempt to overcome catastrophic forgetting by revisiting exemplar samples from past episodes; these methods sample an exemplar set from each episode, which is appended to the data in the following episode. In each episode, the deep model is trained using the data from that episode, as well as the exemplars obtained from previous episodes [1, 8]. Several strategies have been explored to select the exemplar set, such as *Rainbow Memory* [3] which selects uncertain samples given by the variance of the model predictions on a pertubed set of samples; *GDumb* [31] which greedily selects samples with the constraint to asymptotically balance the class distribution of the selected samples; and *iCaRL* [34] which iteratively con-

structs the exemplar set such that the newly added exemplar causes the average feature vector over all the exemplars to best approximate the average feature vector over all the training examples. Generative replay based methods have also been studied, which attempt to model the distribution of data in former episodes and generate samples accordingly [14, 39]. Model-centric methods either regularize the model parameters or expand the network structure for stronger representation ability [13, 43]. Algorithm-centric CIL methods focus on designing algorithms to maintain the model's knowledge in former tasks to mitigate catastrophic forgetting. Knowledge distillation-based CIL methods fall in this category, where the goal is to build a mapping between old and new models and transfer the knowledge of the old model in inducing the new model [24, 49, 50].

Active Learning: Active Learning (AL) is a well-studied problem in the machine vision literature [38]. With the popularity of deep neural networks, *deep active learning (DAL)* has attracted research attention, where the goal is to query informative unlabeled samples for manual annotation and simultaneously learn discriminating feature representations using a deep neural network [36]. Common DAL techniques include a task agnostic scheme which learns a loss prediction function to predict the loss value of an unlabeled sample and queries samples accordingly [45], a greedy technique to query a coreset of samples that represent the whole dataset [37], a sampling technique based on diverse gradient embeddings *(BADGE)* [2], a strategy based on temporal output discrepancy that queries samples based on the discrepancy of outputs given by the models at different optimization steps during training [18] and an AL framework that queries unlabeled samples that can provide the most positive influence on model performance [26]. Techniques based on adversarial learning have depicted particularly impressive performance in DAL [10, 41, 48].

Even though both CIL and AL have been well-researched, their combination has received significantly less attention. A body of research has focused on updating the deep model on-the-fly (rather than retraining from scratch) with actively sampled training data [6]; this setup assumes that the entire training and unlabeled sets are accessible to the deep model throughout the process, which is different from the classical CIL setting. Active class selection has also been studied in the context of CIL, where the incremental learner selects the classes to receive additional training instances from [27]; this is also different from the classical CIL setting, where the learner has no control over the classes of the samples arriving in each episode. Belouadah *et al*. studied the performance of AL algorithms in a CIL setup [4]. However, *ACIL* differs from [4] in the following two fundamental ways: (i) [4] assumes all the samples are annotated in the first episode, which may restrict its application; *ACIL* does not make this assumption and is thus more realistic. (ii) The active sampling criterion in [4] is applied only on the unlabeled data in each episode to create the exemplar set; in contrast, *ACIL* samples intelligently from the unlabeled set and also the exemplar set obtained from the previous episode, and thus addresses catastrophic forgetting more efficiently. This has been empirically validated in Section 4, where *ACIL* has been compared against this method with two state-of-the-art AL techniques: *Coreset* and *BADGE*.

3 Proposed Framework

3.1 Problem Setup

The setup of *ACIL* is depicted in Figure 1. The training data arrives sequentially over N episodes (numbered $0, 1, \ldots N-1$); we consider the *disjoint CIL* setup [3], where the sam-

ples in a given episode belong to different classes compared to the samples in all the previous episodes. The data in a particular episode n consists of two parts: a labeled part X_n^L and an unlabeled part X_n^U ($|X_n^L| \ll |X_n^U|$). Except episode 0, each episode also contains a labeled exemplar set E_{n-1}, which contains a subset of samples from all the previous episodes. From each episode n, we are allowed to select an exemplar set E_n of a fixed size k, containing a subset of samples from X_n^L, X_n^U and E_{n-1}. The samples in the selected exemplar set E_n are annotated and appended to the data in the next episode $(n+1)$. Note that, since X_n^L and E_{n-1} are already annotated, annotating the samples in the selected exemplar set E_n amounts to annotating only the samples selected from X_n^U. In each episode e, a deep neural network is trained on the labeled data X_n^L and E_{n-1}. Note that the size (budget) of the exemplar set remains constant throughout the learning process and does not increase with increasing number of episodes; this appropriately captures a real-world learning setup where the memory budget is fixed. *Also note that the learning setup in ACIL is similar in spirit to replay based methods in data-centric CIL; however, these methods require all the samples to be annotated in each episode, which entails significant manual labor. In contrast, ACIL identifies an exemplar set in each episode without requiring all the samples to be annotated, and thus results in substantial savings in annotation effort.*

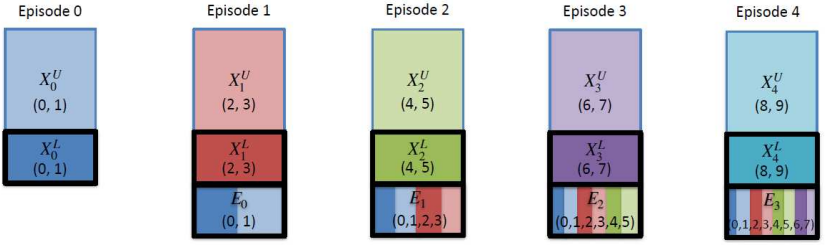


Figure 1: Learning setup of *ACIL*. The numbers in parentheses denote the class labels present in the corresponding set. We consider the disjoint CIL setup [3], where the samples is a given episode belong to different classes compared to the samples in all the previous episodes. Please refer to the text for more details.

In the following sections, we detail how we split the budget k to select samples from the sets X_n^L, X_n^U and E_{n-1} in a given episode n, our strategy to actively select the samples from each set and the loss function used to train the deep neural network in each episode.

3.2 Budget Splitting Strategy

Conventional AL algorithms select all the samples only from the unlabeled set. However, in our setup, selecting all the k samples from the unlabeled set will only capture information about the data in the current episode; this will fail to capture any knowledge about the data in former episodes, which may result in catastrophic forgetting. We therefore also need to select samples from the exemplar set E_{n-1}, as it contains useful information about the data encountered in the previous episodes. Hence, we propose a split of the allowed budget k to select samples both from the labeled exemplar set and the unlabeled set in each episode.

Let $C_{exemplar}$ denote the set of classes in the exemplar set E_{n-1}; this can be easily obtained since the exemplar set is fully labeled. Also, let $C_{episode}$ denote the set of classes that are contained in the current episodic data; this can be obtained since a part of the episodic data X_n^L is labeled (we assume X_n^L contains at least one sample from each class that is present in

the current episodic data). Our rationale is to split the budget k between X_n^U and E_{n-1} in the same proportion as the number of classes in the two sets. Thus, the number of samples to be selected from the episodic unlabeled set X_n^U is:

$$k_{unlabeled} = \left(\frac{|C_{episode}|}{|C_{episode}| + |C_{exemplar}|} \right) .k \tag{1}$$

Similarly, the number of samples to be selected from the episodic exemplar set E_{n-1} is:

$$k_{exemplar} = \left(\frac{|C_{exemplar}|}{|C_{episode}| + |C_{exemplar}|} \right) .k \tag{2}$$

Such a split ensures that we have more budget allocated for the set that contains more classes. Further, our active sampling strategy (detailed below) is applied class-wise on X_n^U and E_{n-1}. Thus, the budget for each class in X_n^U is $\frac{k_{unlabeled}}{|C_{episode}|}$ and that for each class in E_{n-1} is $\frac{k_{exemplar}}{|C_{exemplar}|}$. Since the samples in X_n^U are unlabeled, we use the pseudo-labels furnished by the deep neural network obtained in episode $n-1$ to select samples from each class in X_n^U. Such a budget allocation strategy ensures that the exemplar set E_n selected from episode n contains a good representation of all the classes seen so far; thus, when it is appended to the data in episode $(n+1)$, the deep model trained in episode $(n+1)$ will potentially mitigate catastrophic forgetting. We do not select any samples from X_n^L in E_n, as X_n^L contains the same classes as X_n^U and since $|X_n^U| \gg |X_n^L|$, it most likely contains more informative samples than X_n^L. Thus, the budget for the current episode is completely allocated to X_n^U.

3.3 Active Sampling Strategy

We use the same active sampling strategy to select samples from each class separately in the sets X_n^U and E_{n-1}. In this section, we present the general algorithm to actively sample a subset of a given size from a given set of samples. Let $X = \{x_1, x_2, \ldots x_L\}$ denote a set of L samples, from which we are tasked to select a batch of B samples in our exemplar set. Let $\mathcal{F}(x_i)$ denote the feature embedding of sample x_i, obtained from the trained DNN. We exploit a strategy based on uncertainty and diversity to select the B samples. Specifically, we identify the diverse samples by partitioning the set X into B diverse sets using a partition function $\mathcal{P} : X \rightarrow \{X_1, X_2, \ldots X_B\}$. Let $\{c_1, c_2, \ldots c_B\}$ denote the corresponding centroid of each set. Our objective is to group similar instances in the feature space into a set X_b, $b = 1, 2, \ldots B$, that is, each partition X_b should have low variance $\sigma^2(X_b)$. Using the difference method [46], the variance $\sigma^2(X_b)$ of partition X_b can be computed as:

$$\sigma^2(X_b) = \frac{1}{2|X_b|^2} \sum_{x_i, x_j \in X_b} ||\mathcal{F}(x_i) - \mathcal{F}(x_j)||^2 \tag{3}$$

To incorporate the prediction uncertainty in sample selection, we reformulate the set partitioning problem to minimize the *weighted variance* within each partition (instead of the simple variance), where each sample is weighted based on its prediction uncertainty [33]. The objective function for set partitioning using weighted variance can be posed as:

$$\arg\min_{\mathcal{P}} \sum_{b=1}^{B} \sum_{x \in X_b} \mathcal{I}(x) ||\mathcal{F}(x) - c_b||^2 \tag{4}$$

where $\mathcal{I}(x)$ denotes the prediction uncertainty of sample x, computed using the Shannon's entropy as:

$$\mathcal{I}(x) = -\sum_{i=1}^{C} p(x)_i \log p(x)_i \tag{5}$$

where $p(x)_i$ denotes the probability of sample x with respect to class i and C is the total number of classes. Entropy is a widely used measure of uncertainty in AL research [36].

The optimization problem in Equation (4) in NP-hard. We use the weighted k-means algorithm [25] to approximate the solution to this problem, where k is taken to be equal to the budget B. After clustering, the sample closest to the weighted mean in each partition X_b is selected in the exemplar set. Weighted k-means has been used to solve set partitioning problems with promising results [5, 32].

3.4 Loss Function

In each episode n, a deep neural network is trained on the labeled episodic data X_n^L and the labeled exemplar set E_{n-1} obtained from the previous episode. Since typically, $|X_n^L| \gg |E_{n-1}|$, there may exist a class imbalance in the labeled set $X_n^L \cup E_{n-1}$. We therefore use a weighted cross entropy loss [17] on the labeled set; for a sample x_i in the labeled set, the loss is computed as:

$$\mathcal{L}_{\mathcal{WCE}}(x_i) = -\sum_{j=1}^{C} \delta(y_i == j) w_j \log p_{ij} \tag{6}$$

Here $\delta(.)$ is the indicator function (whose value is 1 if the argument is true and 0 otherwise), p_{ij} denotes the probability of sample x_i with respect to class j obtained using the soft-max activation of the DNN. w_j denotes the importance weight assigned to class j and is inversely proportional to the number of samples in class j in the labeled set: $w_j = \frac{\alpha}{n_j}$, where n_j is the number of samples in class j. The total loss over all the labeled samples is given by:

$$\mathcal{L}_{\mathcal{WCE}} = \frac{1}{|L|} \sum_{i=1}^{|L|} \mathcal{L}_{\mathcal{WCE}}(x_i) \tag{7}$$

Further, knowledge distillation has proved to be effective in CIL, where knowledge from the deep model $\mathcal{M}_{n-1}$, trained in episode $n-1$, is distilled to train the deep model $\mathcal{M}_n$ in episode n. $\mathcal{M}_{n-1}$ is applied on the exemplar set E_{n-1} and a distillation loss $\mathcal{L}_{\mathcal{D}}$ is computed between the predictions furnished by $\mathcal{M}_n$ and $\mathcal{M}_{n-1}$ on E_{n-1}. Please refer to [7, 35] for details about computing the distillation loss in a CIL setting. The total loss to train the DNN in episode n is thus:

$$\mathcal{L} = \mathcal{L}_{\mathcal{WCE}} + \lambda \mathcal{L}_{\mathcal{D}} \tag{8}$$

where λ is a weight parameter controlling the relative importance of the two terms. The pseudo-code of *ACIL* is provided in the Supplemental File. As evident from the pseudo-code, our algorithm is computationally lightweight, simple and easy to implement.

4 Experiments and Results

Datasets: We used six computer vision datasets to study the performance of *ACIL*: MNIST [9], SVHN [29], CIFAR 10 [21], CIFAR 100 [21], COIL [28] and Tiny ImageNet

[23]. The sizes of the labeled set X_n^L, unlabeled set X_n^U and exemplar set E_n for each dataset are detailed in the Supplemental file.

Comparison Baselines: We used methods from both IL and AL as comparison baselines in our work: (i) **AL baselines:** We used 3 active learning techniques as comparison baselines: *Random Sampling*, which queries a batch of samples at random from the current episode; *Coreset* [37]; and *BADGE* [2]. Both *Coreset* and *BADGE* are widely used AL algorithms [36]. Similar to *ACIL*, the AL baselines do not require all the samples to be annotated in each episode and operate on X_n^L and X_n^U. As proposed in [4], the samples in the exemplar set are selected from the unlabeled data only in each episode, by applying different AL techniques. (ii) **CIL baselines:** We used the following CIL methods as baselines in our research: *iCaRL* [34]; *GDumb* [31]; *Rainbow Memory* [3]; and *Finetuning* [47]. Note that, *iCaRL*, *GDumb* and *Rainbow Memory* are all recently proposed replay based CIL techniques which use an exemplar set in each episode to mitigate catastrophic forgetting; these were selected since *ACIL* closely resembles a replay based CIL setting. *Finetuning* is a control CIL baseline, which does not use an exemplar set, instead it uses the same model that was trained on the previous episode and incrementally updates it using the data in the current episode.

Evaluation Metrics: We used the following metrics to evaluate the performance of our framework: (i) *Accuracy*, which depicts how well the deep model generalizes on test data containing classes seen in the current and all the previous episodes; after each episode n, this metric was computed as the average accuracy of the model on the test set of each episode $1, 2, \ldots, n$. (ii) *Annotation effort*, computed as the number of samples that had to be annotated in each episode. (iii) *Retention* (a metric for forgetting), which quantifies how much information the model retains from the first episode, and was computed as the accuracy of the model on the test set from the first episode. Accuracy and retention are commonly used metrics in CIL [47] while human annotation effort is used to quantify the performance of AL algorithms [36].

Implementation Details: We used ResNet-34 [15] as the backbone architecture in our work. The model was trained for 240 epochs in each episode, with a batch size of 128 and a learning rate of 0.001, using the *Adam* optimizer.

4.1 Main Results

The learning performance results are depicted in Figure 2 (accuracy) and Table 1 (annotation effort). In each graph in Figure 2, the x-axis denotes the episode number, and the y-axis denotes the accuracy. Each plot represents the mean results of three runs to rule out the effects of randomness. Since the deep model is continually updated with new class information, the accuracy decays as more classes are incorporated with increasing episodes.

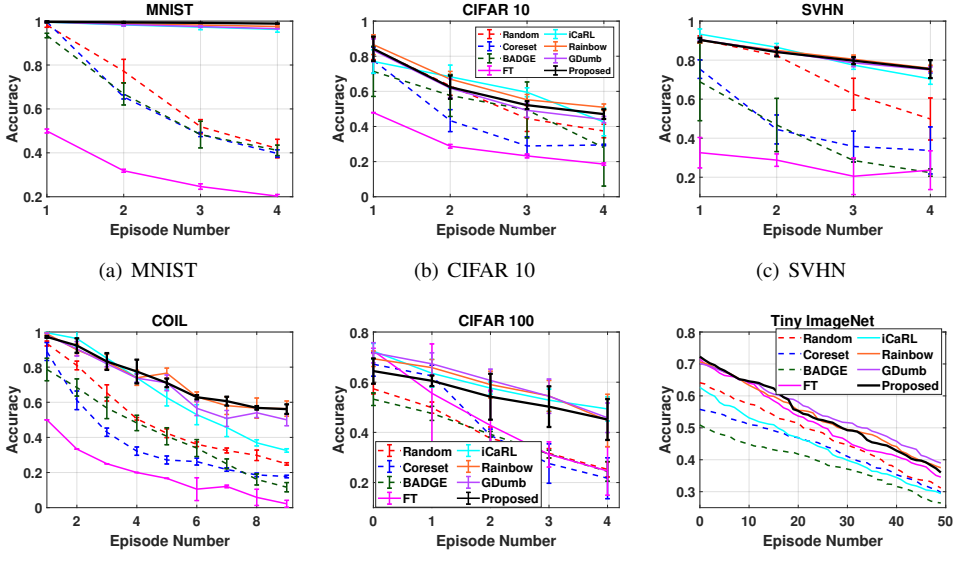
Accuracy: *ACIL* vs. AL baselines: *ACIL* comprehensively outperforms all the AL baselines, and depicts much better accuracy. This is because, even though the AL baselines select an exemplar set in each episode, the samples are selected completely from the unlabeled set of the corresponding episode, as proposed in [4]; thus, they fail to capture the knowledge from the former episodes and hence are not effective in mitigating catastrophic forgetting.

Accuracy: *ACIL* vs. CIL baselines: *ACIL* depicts comparable performance as the CIL baselines, *iCaRL*, *GDumb* and *Rainbow*. 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 accuracy drops at more or less the same rate as the CIL baselines, with increasing

number of episodes. *Finetuning* depicts much worse performance, as it does not use an exemplar set to retain knowledge from previous episodes.

We note that *ACIL* sometimes depicts slightly better learning performance than the CIL baselines (where all the samples are labeled in all the episodes); for instance, it is better than *iCaRL* for Tiny ImageNet. *iCaRL* selects the exemplar set iteratively such that the mean feature vector of the exemplar is a good approximation of the mean feature vector of all the training samples in a given episode. This does not necessarily guarantee a good set of exemplars. For instance, it is possible to select a batch of uninformative / redundant samples whose mean is very close to the mean of all the samples. *ACIL* on the other hand, selects samples by optimizing a criterion based on uncertainty and diversity, and thus selects a better set of exemplars to append to the next episode. Thus, even though all the samples are labeled in all the episodes in *iCaRL*, by selecting a better set of exemplars to propagate from episode to episode, *ACIL* is able to obtain a better learning performance.

ACIL also performs better than *GDumb* in the later episodes of CIFAR 10 and COIL. *GDumb* selects the exemplars to ensure that the class distributions are balanced; it does not consider the quality / informativeness of the exemplars. *ACIL* propagates a more informative set of exemplars to the subsequent episode, which explains its marginally better performance.



(a) MNIST (b) CIFAR 10 (c) SVHN

(d) COIL (e) CIFAR 100 (f) Tiny ImageNet

Figure 2: Performance analysis of *ACIL*. 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.

Annotation effort: Table 1 reports the average number of samples that had to be annotated per episode (including the episodic labeled set X_n^L) for all the methods. The CIL baselines require all the samples to be annotated in each episode, which incurs substantial annotation effort. The AL baselines depict much less annotation effort, as only k unlabeled samples need to be annotated in each episode. *ACIL* depicts the least annotation effort, as it splits the budget k to select samples from the exemplar set E_{n-1} and the unlabeled set X_n^U in

	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 1: 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*).

episode n; since the exemplar set is already annotated, only the samples selected from X_n^U need to be annotated in each episode for *ACIL*. For the CIFAR 10 dataset, for instance, *ACIL* results in about 4.13 fold reduction in annotation effort on average in each episode, compared to the CIL baselines, while furnishing performance very similar to these baselines. Thus, *ACIL* depicts comparable (and sometimes marginally better) performance than the CIL baselines at substantially reduced annotation effort; it also depicts much better performance than the AL baselines, at slightly reduced annotation effort. *ACIL* is thus a promising solution for real-world incremental learning applications.

	Random	Coreset	BADGE	FT	iCaRL	Rainbow	GDumb	Proposed
MNIST	0.41 ± 0.03	0.39 ± 0.01	0.41 ± 0.02	0.20 ± 0.01	0.96 ± 0.01	$\underline{0.97} \pm 0.01$	0.96 ± 0.01	$\mathbf{0.98} \pm 0.02$
CIFAR 10	0.37 ± 0.03	0.29 ± 0.03	0.27 ± 0.17	0.18 ± 0.04	0.42 ± 0.06	$\mathbf{0.5} \pm 0.02$	0.43 ± 0.01	$\underline{0.47} \pm 0.02$
SVHN	0.49 ± 0.09	0.33 ± 0.1	0.22 ± 0.02	0.23 ± 0.08	$\underline{0.70} \pm 0.02$	$\mathbf{0.75} \pm 0.03$	$\mathbf{0.75} \pm 0.01$	$\mathbf{0.75} \pm 0.04$
COIL	0.24 ± 0.05	0.17 ± 0.02	0.11 ± 0.02	0.02 ± 0.02	0.32 ± 0.01	$\mathbf{0.56} \pm 0.04$	$\underline{0.5} \pm 0.03$	$\mathbf{0.56} \pm 0.02$
CIFAR 100	0.25 ± 0.03	0.22 ± 0.08	0.24 ± 0.04	0.25 ± 0.10	$\mathbf{0.49} \pm 0.05$	0.45 ± 0.10	$\underline{0.46} \pm 0.06$	0.45 ± 0.08
Tiny ImageNet	0.31 ± 0.01	0.30 ± 0.03	0.26 ± 0.04	0.35 ± 0.02	0.29 ± 0.02	$\underline{0.38} \pm 0.05$	$\mathbf{0.39} \pm 0.04$	0.36 ± 0.04

Table 2: Average ($\pm$ std) accuracy after the last episode achieved by the CIL baselines (*Finetuning, iCaRL, Rainbow, GDumb*), AL baselines (*Random, Coreset, BADGE*) and the proposed method (*ACIL*). Best results are marked in **bold** and second best results are underlined.

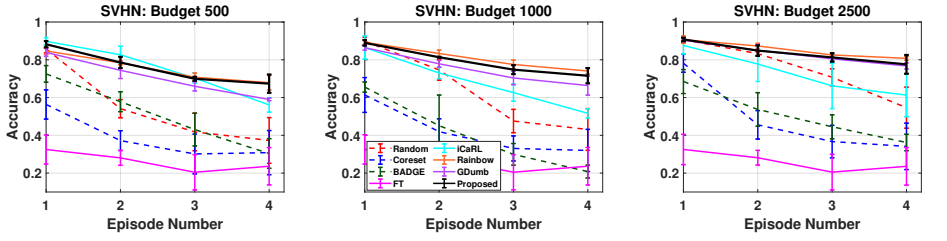
Table 2 reports the accuracy obtained after the last episode for all the methods. We note that the proposed method achieves the highest accuracy in 3 out of the 6 datasets and the second highest accuracy in 1 dataset. These results further corroborate the fact that *ACIL* depicts competitive performance as the CIL baselines, at much reduced annotation effort.

4.2 Study of the exemplar set size (query budget)

The goal of this experiment was to study the effect of the exemplar set size (budget) on the learning performance. The results on the SVHN dataset with budgets 500, 1,000 and 2,500 are depicted in Figure 3 (accuracy) and Table 3 (annotation effort) (the default budget in Figure 2(c) was 2,000). We note that the accuracy shows an increasing trend with an increase in budget, which is intuitive. A similar pattern is evident for all the budgets, where *ACIL* depicts accuracy comparable to the CIL baselines (*iCaRL*, *GDumb* and *Rainbow*) and much better than the AL baselines.

From Table 3, it is also evident that *ACIL* results in substantial savings in the human annotation effort compared to the CIL methods. The CIL baselines need all the samples to be annotated in all the episodes, so their annotation cost is not affected by the budget of the exemplar set. *ACIL*, on the other hand, distributes the available budget between the unlabeled samples in a given episode and the exemplar set obtained from the previous episode. The results demonstrate that *ACIL* depicts comparable performance to the CIL baselines at substantially reduced human annotation effort; it depicts much better performance than the AL baselines at slightly reduced annotation effort. These results depict the robustness of

ACIL to the budget of the exemplar set. This result is particularly important from a practical standpoint, since the budget of the exemplar set is dependent on the available memory / resources of a given application, and is different for different applications.



(a) Budget 500 (b) Budget 1000 (c) Budget 2500

Figure 3: Study of the exemplar set size (budget) on the SVHN dataset. 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. Best viewed in color.

	CIL Baselines	AL Baselines	Proposed
SVHN: Budget 500	$14,651.40 \pm 5,542.22$	$3,330 \pm 1,068.78$	$3,139.20 \pm 1,166.54$
SVHN: Budget 1,000	$14,651.40 \pm 5,542.22$	$3,730 \pm 1,068.41$	$3,347.20 \pm 1,246.61$
SVHN: Budget 2,500	$14,651.40 \pm 5,542.22$	$4,930 \pm 1,285.86$	$3,972.80 \pm 1,577.21$

Table 3: 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*) for different budgets of the exemplar set on the SVHN dataset

We also conducted the following experiments and the results are reported in the Supplemental File: (i) *study of the backbone network architecture;* (ii) *learning performance with varying number of episodes;* (iii) *an analysis of the computation time of all the methods studied;* (iv) *ablation studies to analyze the importance of the uncertainty and diversity components in our framework; and* (v) *performance analysis of our framework using the retention metric.*

5 Conclusion and Future Work

In this paper, we proposed *ACIL*, a novel active learning framework for class incremental learning settings. Contrary to existing CIL techniques, *ACIL* does not require all the samples in each episode to be annotated; it selects an exemplar set in each episode, which is annotated and appended to the data in the next episode, so that the deep neural network trained in a given episode can retain knowledge of the classes encountered in past episodes. We formulated a criterion based on uncertainty and diversity to select the exemplar set in each episode; we also proposed a budget splitting strategy to ensure the exemplar set captures informative samples from all classes seen in former episodes. Our extensive empirical analyses on six vision datasets corroborated that *ACIL* can not only mitigate catastrophic forgetting and deliver accuracy comparable to state-of-the-art CIL methods, but can also result in substantial savings of human annotation effort. As part of future work, we plan to extend our framework to address IL in the regression setup, which has attracted research attention recently [16].

6 Acknowledgment

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

References

[1] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio. Gradient based sample selection for online continual learning. In *Neural Information Processing Systems (NeurIPS)*, 2019.

[2] J. Ash, C. Zhang, A. Krishnamurthy, J. Langford, and A. Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations (ICLR)*, 2020.

[3] J. Bang, H. Kim, Y. Yoo, J. Ha, and J. Choi. Rainbow Memory: Continual learning with a memory of diverse samples. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[4] 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.

[5] S. Borgwardt, A. Brieden, and P. Gritzmann. A balanced k-means algorithm for weighted point sets. In *arXiv:1308.4004v2*, 2016.

[6] C. Brust, C. Käding, and J. Denzler. Active and incremental learning with weak supervision. *KI-Künstliche Intelligenz*, 34:165–180, 2020.

[7] F. Cermelli, D. Fontanel, A. Tavera, M. Ciccone, and B. Caputo. Incremental learning in semantic segmentation from image labels. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[8] A. Chaudhry, P. Dokania, T. Ajanthan, and P. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *European Conference on Computer Vision (ECCV)*, 2018.

[9] L. Deng. The MNIST database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[10] M. Ducoffe and F. Precioso. Adversarial active learning for deep networks: a margin based approach. In *International Conference on Machine Learning (ICML)*, 2018.

[11] L. Floridi and M. Chiriatti. GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30:681–694, 2020.

[12] M. Gaber. Advances in data stream mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):79–85, 2012.

[13] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[14] C. He, R. Wang, S. Shan, and X. Chen. Exemplar-supported generative reproduction for class incremental learning. In *British Machine Vision Conference (BMVC)*, 2018.

[15] 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.

[16] Y. He and B. Sick. Clear: An adaptive continual learning framework for regression tasks. In *arXiv:2101.00926v4*, 2021.

[17] Y. Ho and S. Wookey. The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling. *IEEE Access*, 8:4806 – 4813, 2019.

[18] S. Huang, T. Wang, H. Xiong, J. Huan, and D. Dou. Semi-supervised active learning with temporal output discrepancy. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[19] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A.Žídek, A. Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[20] G. Krempl, I. Žliobaite, D. Brzeziński, E. Hüllermeier, M. Last, V. Lemaire, T. Noack, A. Shaker, S. Sievi, M. Spiliopoulou, et al. Open challenges for data stream mining research. *ACM SIGKDD Explorations Newsletter*, 16(1):1–10, 2014.

[21] A. Krizhevsky. Learning multiple layers of features from tiny images. In *Technical Report, University of Toronto*, 2009.

[22] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44 (7):3366–3385, 2021.

[23] Y. Le and X. Yang. Tiny imagenet visual recognition challenge. In *Technical Report, Stanford University*, 2015.

[24] Z. Li and D. Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(12):2935–2947, 2017.

[25] H. Liu, J. Wu, T. Liu, D. Tao, and Y. Fu. Spectral ensemble clustering via weighted k-means: Theoretical and practical evidence. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 29(5):1129 – 1143, 2017.

[26] Z. Liu, H. Ding, H. Zhong, W. Li, J. Dai, and C. He. Influence selection for active learning. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[27] C. McClurg, A. Ayub, H. Tyagi, S. Rajtmajer, and A. Wagner. Active class selection for few-shot class-incremental learning. In *Conference on Lifelong Learning Agents*, 2023.

[28] S. Nene, S. Nayar, and H. Murase. Columbia object image library (coil-100). In *Technical Report CUCS-006-96*, 1996.

[29] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Ng. Reading digits in natural images with unsupervised feature learning. In *Neural Information Processing Systems (NeurIPS) Workshop*, 2011.

[30] H. Osmanbeyoglu, J. Wehner, J. Carbonell, and M. Ganapathiraju. Active machine learning for transmembrane helix prediction. *BMC Bioinformatics*, 11(1), 2010.

[31] A. Prabhu, P. Torr, and P. Dokania. GDumb: A simple approach that questions our progress in continual learning. In *European Conference on Computer Vision (ECCV)*, 2020.

[32] V. Prabhu, A. Chandrasekaran, K. Saenko, and J. Hoffman. Active domain adaptation via clustering uncertainty-weighted embeddings. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[33] G. Price. Extension of covariance selection mathematics. *Annals of human genetics*, 35(4):485–490, 1972.

[34] S. Rebuffi, A. Kolesnikov, G. Sperl, and C. Lampert. iCaRL: Incremental classifier and representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[35] S. Rebuffi, A. Kolesnikov, G. Sperl, and C. Lampert. iCaRL: Incremental classifier and representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[36] P. Ren, Y. Xiao, X. Chang, P. Huang, Z. Li, B. Gupta, X. Chen, and X. Wang. A survey of deep active learning. *ACM Computing Surveys*, 54(9), 2021.

[37] O. Sener and S. Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations (ICLR)*, 2018.

[38] B. Settles. Active learning literature survey. In *Technical Report 1648, University of Wisconsin-Madison*, 2010.

[39] H. Shin, J. Lee, J. Kim, and J. Kim. Continual learning with deep generative replay. In *Neural Information Processing Systems (NeurIPS)*, 2017.

[40] D. Silver, A. Huang, C. Maddison, A. Guez, L. Sifre, G. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[41] S. Sinha, S. Ebrahimi, and T. Darrell. Variational adversarial active learning. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[42] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research (JMLR)*, 2:45–66, 2001.

[43] Z. Wang, R. Turko, O. Shaikh, H. Park, N. Das, F. Hohman, M. Kahng, and D. Chau. Cnn explainer: Learning convolutional neural networks with interactive visualization. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 27(2):1396–1406, 2020.

[44] X. Wu, C. Chen, M. Zhong, J. Wang, and J. Shi. Covid-al: The diagnosis of covid-19 with deep active learning. *Medical Image Analysis*, 68, 2021.

[45] D. Yoo and I. Kweon. Learning loss for active learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[46] Y. Zhang, H. Wu, and L. Cheng. Some new deformation formulas about variance and covariance. In *IEEE International Conference on Modelling, Identification and Control*, 2012.

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

[48] J. Zhu and J. Bento. Generative adversarial active learning. In *arXiv:1702.07956*, 2017.

[49] Z.Zhou and Y. Jiang. Nec4. 5: Neural ensemble based c4. 5. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 16(6):770–773, 2004.

[50] Z.Zhou, Y. Jiang, and S. Chen. Extracting symbolic rules from trained neural network ensembles. *AI Communications*, 16(1):3–15, 2003.