Supplementary Materials

1 CBCL Algorithms

The algorithms below describe portions of the complete CBCL algorithm. Algorithm 1 is for *Agg-Var* clustering (Section 3.1 in paper), Algorithm 2 is for the weighted voting scheme (Section 3.2 in paper) and Algorithm 3 is for centroid reduction technique (Section 3.3 in paper).

Algorithm 1 CBCL: *Agg-Var* Clustering

Input: $X = \{X^1, ..., X^t\}$ ▷ feature vector sets of the training images belonging to t classes
require: D ▷ distance threshold
Output: $C = \{C^1, ..., C^t\}$ ▷ collection of class centroid sets for t classes

for $j = 1$; $j \leq t$ **do**
  $C^j \leftarrow \{x_1^j\}$ ▷ initialize centroids for each class
for $j = 1$; $j \leq t$ **do**
  for $i = 2$; $i \leq N_j$ **do**
    $d_{min} \leftarrow \min_{l=1,..,size(C^j)} dist(c_l^j, x_i^j)$ ▷ distance to closest centroid
    $i_{min} \leftarrow \text{argmin}_{l=1,..,size(C^j)} dist(c_l^j, x_i^j)$ ▷ index of the closest centroid
    Set $w_{i_{min}}^j$ to be the number of images clustered
    in the i_{min}th centroid pair of class j
    if $d_{min} < D$ **then**
      $c_{i_{min}}^j \leftarrow \frac{w_{i_{min}}^j \times c_{i_{min}}^j + x_i^j}{w_{i_{min}}^j + 1}$ ▷ update the closest centroid
    else
      $C^j.append(x_i^j)$ ▷ add a new centroid for class j

Algorithm 2 CBCL: Weighed voting scheme for classification

Input: x ▷ feature vector of the test image
require: n ▷ number of closest centroids for prediction
require: $C = \{C^1, ..., C^t\}$ ▷ class centroids sets
require: $\{N_1, N_2, ..., N_t\}$ ▷ Number of training images per class
Output: y^* ▷ predicted label

$C^* = \{c_1, c_2, ..., c_n\}$ ▷ n closest centroids from set C
for $y = 1$;$y \leq t$ **do**
  $Pred(y) = \frac{1}{N_y} \sum_{j=1}^{n} \frac{1}{dist(x, c_j)} [y_j = y]$
$y^* = \text{argmax}_{y=1,...,t} Pred(y)$

Algorithm 3 CBCL: Centroid Reduction

Input: $C = \{C^1, ..., C^t\}$ ▷ current class centroids sets
require: K ▷ maximum number of centroids
require: K_{new} ▷ number of centroids for new classes
Output: $C_{new} = \{C^1_{new}, ..., C^t_{new}\}$ ▷ reduced class centroids sets

```
K_r = K_t + K_new − K
for y = 1; y ≤ t do
    N*_y(new) = N*_y(1 − K_r/K_t)
    C^y_new = k-means(n_clusters = N*_y(new), C^y)
```

2 Comparison of CBCL with FearNet on CIFAR-100 Dataset

In this section we compare CBCL against FearNet [1] which is another brain-inspired model for incremental learning. FearNet uses a ResNet-50 pre-trained on ImageNet for feature extraction and uses brain-inspired dual-memory model. FearNet stores the feature vectors and covariance matrices for old class images and also uses a generative model for data augmentation. For this comparison we use the evaluation metrics provided in [1]. We test the model's ability to retain base-knowledge given as $\Omega_{base} = \frac{1}{T-1}\sum_{t=2}^{T} \frac{\alpha_{base,t}}{\alpha_{offline}}$, where $\alpha_{base,t}$ is the accuracy of the model on the classes learned in the first increment, $\alpha_{offline}$ is the accuracy of a multi-layer perceptron trained offline (69.9% reported in [1]) and T is the total number of increments. The model's ability to recall new information is evaluated as $\Omega_{new} = \frac{1}{T-1}\sum_{t=2}^{T} \alpha_{new,t}$, where $\alpha_{new,t}$ is the accuracy of the model on the classes learned in increment t. Lastly, we evaluate the model on all test data as $\Omega_{all} = \frac{1}{T-1}\sum_{t=2}^{T} \frac{\alpha_{all,t}}{\alpha_{offline}}$, where $\alpha_{all,t}$ is the accuracy of the model on all the classes learned up till increment t. For a fair comparison, we use the ResNet-50 pre-trained on ImageNet as a feature extractor.

Evaluation Metric	**FearNet**	**CBCL**	**CBCL 5-Shot**	**CBCL 10-Shot**
Ω_{base}	0.927	**1.025**	0.754	0.830
Ω_{new}	0.824	**1.020**	0.778	**0.870**
Ω_{all}	0.947	**1.025**	0.778	0.870

Table 1: Comparison with FearNet on CIFAR-100. Ω_{base}, Ω_{new} and Ω_{all} are all normalized by the offline multi-layer preceptron (MLP) baseline (69.9%) reported in [1]. A value greater than 1 means that the *average incremental accuracy* of the model is higher than the offline MLP.

Table 1 compares CBCL with FearNet on CIFAR-100 dataset using the metrics proposed in [1]. We report results of CBCL on the most difficult increment setting (2 base classes and then 1 class per increment for 98 classes) for this experiment. CBCL clearly outperforms FearNet on all three metrics (Ω_{base}, Ω_{new}, Ω_{all}) by a significant margin when using all training examples per class. For 10-shot incremental learning, CBCL outperforms FearNet (which uses all the training examples per class) on Ω_{new} but for Ω_{base} and Ω_{all} it is slightly inferior. For 5-shot incremental learning setting, the results of CBCL are inferior to FearNet (which uses all the training examples) but the change in accuracy is not drastic. It should be noted that even for 10-shot and 5-shot incremental learning settings, the MLP baseline, used during the calculation of Ω_{base}, Ω_{new} and Ω_{all}, has been trained on all the training data of each class in a single batch.

We also trained a ResNet-50 for 5-shot and 10-shot learning with all the class training data available in one batch and the test accuracies for 5-shot and 10-shot learning were 8.49% and 12.21%, respectively. CBCL outperforms this baseline by a remarkable margin for both 5-shot and 10-shot settings, demonstrating that it is extremely effective for few-shot incremental learning setting.

3 Analysis of Different Memory Budgets

We perform a set of experiments on CIFAR-100 dataset to analyze the effect of different memory budgets on the performance of CBCL. We performed these experiments on *hybrid5* as well to show the contribution of our proposed centroid reduction technique towards CBCL's performance. Figure 1 compares the average incremental accuracy of CBCL and *hybrid5* for different memory budgets. As expected, both CBCL and *hybrid5* achieve higher accuracy for when provided higher memory budgets. Furthermore, CBCL constantly outperforms *hybrid5* for all different memory budgets (except for K=9000 when there is no need for any reduction) and the performance gap increases for smaller memory budgets. This clearly shows the effectiveness of our proposed centroid reduction technique over simple removal of centroids. Furthermore, it should be noted that even for only $K = 3000$ centroids CBCL's average incremental accuracy (**67.5%**) is higher than that of the state-of-the-art methods ([2]: 64.84%).

Figure 1: Average incremental accuracy of CBCL and *hybrid5* for different memory budgets (K). The difference between CBCL and *hybrid5* is more prominent for smaller memory budgets.

4 Confusion Matrices

We further provide insight into the behavior of CBCL through the confusion matrix. Figure 2 shows the confusion matrix of CBCL on CIFAR-100 dataset when learning with 10 classes per increment with a memory budget of K =7500. The pattern is quite obvious that the confusion matrix of CBCL looks homogenous in terms of diagonal and off-diagonal entries depicting that CBCL does not get biased towards new or old classes and it does not suffer from *catastrophic forgetting*.

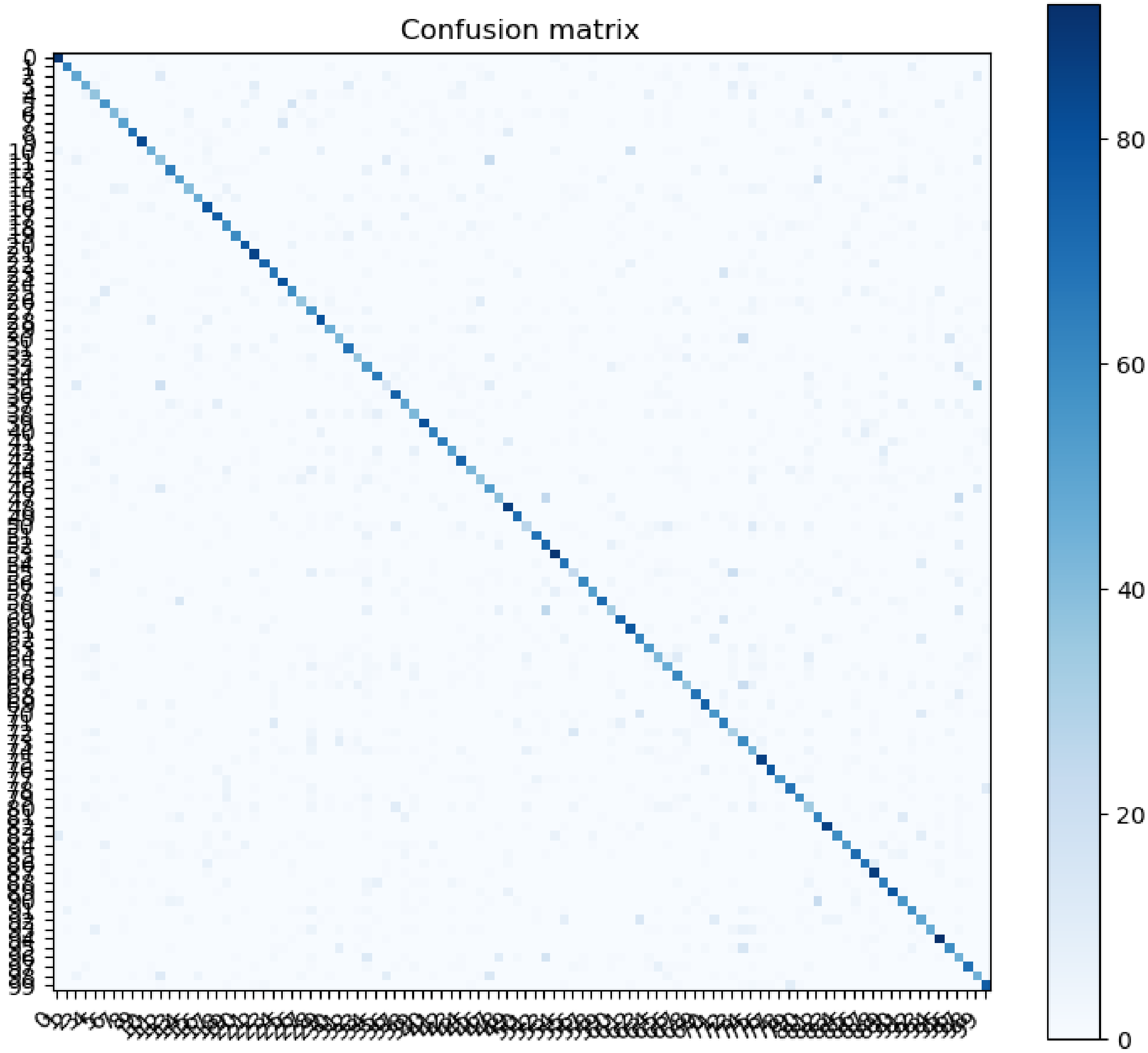

Figure 2: Confusion matrix of CBCL on CIFAR-100 dataset with 10 classes per increment and total centroids limit of K =7500. The vertical axis depeicts the ground truth and the horizontal axis shows the predicted labels (0-99).

5 Hyperparameters

CBCL only has two hyperparameters: distance threshold (D) and number of centroids used for classification (n). For all three datasets (CIFAR-100, Catltech-101 and CUBS-200-2011), D was tuned to one of the values in the set $\{70, 75, 80, 85, 90\}$, although in most of the increments it was tuned to 70 for both incremental learning and FSIL experiments. n was tuned to one of the values in the set $\{1, 2, ..., 10\}$ for incremental learning experiments but for FSIL experiments it was mostly tuned to 1.

6 Results on Caltech-101 Using Bag of Visual Words

To show the effect of feature extractor choice on CBCL's performance, we report results on Caltech-101 dataset using bag of visual words (with SURF features [3]). Bag of visual words (BoVW) features are significantly inferior to CNN features on image classification tasks. Table 2 compares CBCL using BoVW against LWM and *finetuning* (FT) with 10 classes per increment. CBCL's accuracy is significantly lower than LWM and FT for the first increment (because of inferior features) and for all the other 9 increments it is either higher or slightly inferior to LWM. This shows that CBCL yields near state-of-the-art accuracy even when using inferior features. Furthermore, it should be noted that the decrease in accuracy of CBCL is still only **37.61%** after 10 increments while for LWM and FT the decrease in accuracies are 69.52% and and 49.36%. These results clearly show the effectiveness of CBCL to avoid *catastrophic forgetting*.

# Classes	**FT**	**LWM**	**CBCL BoVW**
10 (base)	97.78	97.78	85.14 ± 1.12
20	59.55	75.34	$\mathbf{77.84 \pm 1.86}$
30	52.65	71.78	69.65 ± 2.21
40	44.51	67.49	63.89 ± 1.40
50	35.52	59.79	$\mathbf{60.30 \pm 1.73}$
60	31.18	56.62	$\mathbf{57.20 \pm 1.37}$
70	32.99	54.62	$\mathbf{55.25 \pm 0.99}$
80	27.45	48.71	$\mathbf{51.17 \pm 0.84}$
90	28.55	46.21	$\mathbf{48.13 \pm 0.88}$
100	28.26	48.42	47.53 ± 0.69

Table 2: Comparison with FT and LWM [4] on Caltech-101 dataset in terms of classification accuracy (%) with 10 classes per increment. Average and standard deviation of classification accuracies per increment are reported

References

[1] Ronald Kemker and Christopher Kanan. Fearnet: Brain-inspired model for incremental learning. In *International Conference on Learning Representations*, 2018.

[2] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[3] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008.

[4] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.