Supplementary Materials

1 CORe-50 Dataset

CORe-50 is an object recognition dataset that was developed by Lomonaco et al. [1] for continual learning that contains 50 household objects belonging to 10 classes (5 objects per class). Classification can be performed at the object level (50 classes) or category level (10 classes). Each object class is captured with 11 different background environments from 11 different recording sessions. Each session (video recording) is composed of an approximately 15 second video clip recorded at 20 fps. We used the cropped 128×128 images with sessions 3, 7, and 10 as the test set with the rest being used for training. We performed experiments at the category level with 10 classes.

2 Comparison with AL and CL Setups

For FoCAL experiments on the CORe-50 dataset, we develop a new experimental setup, as the standard active learning and continual learning experimental setups are not sufficient to test FoCAL. In a standard active learning (AL) experiment [2], the training set of a large dataset (such as CIFAR-10 [3]) is divided into smaller subsets ($\sim$10000 unlabeled images), where each subset can contain images belonging to all the classes in the dataset. In a single increment, an active learning model is provided with a subset and the model chooses k($\sim$1000) instances to be labeled. After each increment, the model is tested on the complete test set of the dataset. Note that in subsequent increments, the model still has access to the previous data (batch learning). On the other hand, in a standard continual learning (CL) experiment [4, 5], a large dataset is divided into smaller subsets where each subset contains complete training set of a subset of the classes. In a single increment, a continual learning model is provided with a subset with the ground truth labels. After training in each increment, the model is tested only on the test set of the classes learned so far. Also, in the subsequent increments, the model does not have access to the previous data subsets.

In contrast, for FoCAL experiments on CORe-50 we combined all 8 training sessions in CORe-50 to generate 400 training object instances. In each increment, we randomly sampled $m = 5$ object instances (from 400 instances) and allowed the model to learn the label of $k = 1$ out of 5 object instances, making it a challenging problem to learn from a single object instance in each increment. Once an object was learned by the model, it was removed from the complete set of objects and, thus, was not available to the model in later increments. Hence, we allowed the model to learn all objects in 400 increments with one object learned in each increment.

3 SOTA CL Approaches

We compare our approach against 6 continual learning approaches (LWF [6], EWC [7], CWR [1], iCaRL [5], NCM [8], CBCL [4]), finetuning (FT) and a few-shot batch learning baseline (FLB) [9]. FLB uses a pre-trained CNN to extract feature vectors for images and trains a linear classifier using cross entropy loss. FLB is trained on the complete training data of the previous increments and the current increment. In other words, FLB does not learn continually and therefore should have an advantage over other continual learning approaches (including ours). FT uses the same architecture as FLB but FT is trained only on the data of the current increment. FT suffers from catastrophic forgetting and therefore should produce lower accuracy than other continual learning approaches. Nearest Class Mean (NCM) classifier computes a single centroid for each class as the mean of all

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

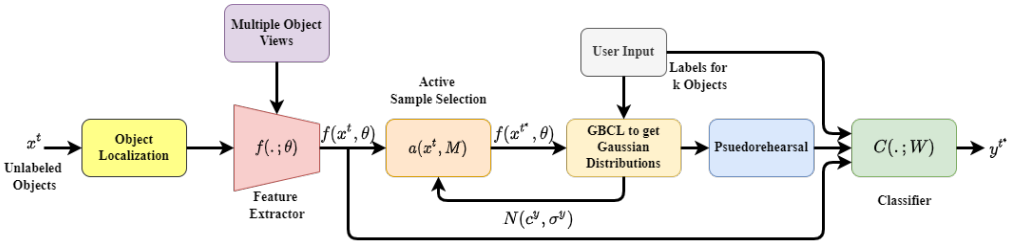

Figure 1: Our overall framework for FoCAL. In each increment t, the features extracted for unlabeled objects $f(x^t;\theta)$ are passed through the acquisition function $a(x^t,\mathcal{M})$ to get k most informative samples x^{t^*}, which are labeled by the oracle. The labeled feature vectors are used to update the GMM representation of the learned classes Y^t. Pseudo-rehearsal is used to replay old class data, and the classifier model $C(.;W)$ is trained on the pseudo-samples of the old classes and the labeled feature vectors in the tth increment.

the feature vectors of the images in the training set for each class. To predict the label for a test image, NCM assigns it the class label of the closest centroid. NCM avoids catastrophic forgetting by using centroids. Each class centroid is computed using only the training data of that class, hence even if the classes are learned continually, the centroids for previous classes are not affected when new classes are learned. Centroid-Based Concept Learning (CBCL) generalizes NCM, and uses a cognitively-inspired clustering approach to learn multiple centroids per class, instead of a single centroid. CBCL uses a k-nearest centroids approach for classification of test images. Incremental classifier and representation learning (iCaRL) [5] combines knowledge distillation [10] and NCM for class-incremental learning. Knowledge distillation uses a distillation loss term that forces the labels of the training data of previously learned classes to remain the same when learning new classes. iCaRL uses the old class data while learning a representation for new classes and uses the NCM classifier for classification of the old and new classes. Elastic Weight Consolidation (EWC) searchs for importance weights for each parameter in the neural network for the old classes. During learning of new classes, the gradients for the importance weights are penalized to prevent a large change in importance parameters. This way the model can mitigate catastrophic forgetting. Finally, CWR uses a similar architecture as in FLB, but it does not use the data of the old classes when leanring new classes. To mitigate catastrophic forgetting, CWR keeps the weights of the linear layer in the classifier from the previous increment, and imprints them in the updated linear layer for learning new classes.

4 FoCAL System on a Humnoid Robot

For real world robotics applications, an autonomous robot might not have perfect images of individual objects available. In such cases, the robot must detect and localize individual objects, capture multiple views of the objects itself, and then ask its human user to provide the label for the most informative objects. Therefore, we develop a complete system for our FoCAL framework (see Figure 1) that can be integrated on a real robot. The subsections below describe the different components of our framework:

4.1 Object Localization

In a real-world environment, the robot might encounter multiple objects at a single location. Before finding the uncertainty scores for unlabeled objects, the robot must first detect, localize, and get images of individual objects from a set of objects (see Figure 2 (a, b) for examples). In our framework, the robot uses RetinaNet [11] for object detection and localization. This network proposes image regions likely to contain objects. After passing the image through the RetinaNet, the locations of the objects are passed on to the manipulation and image capturing module to get images of individual objects.

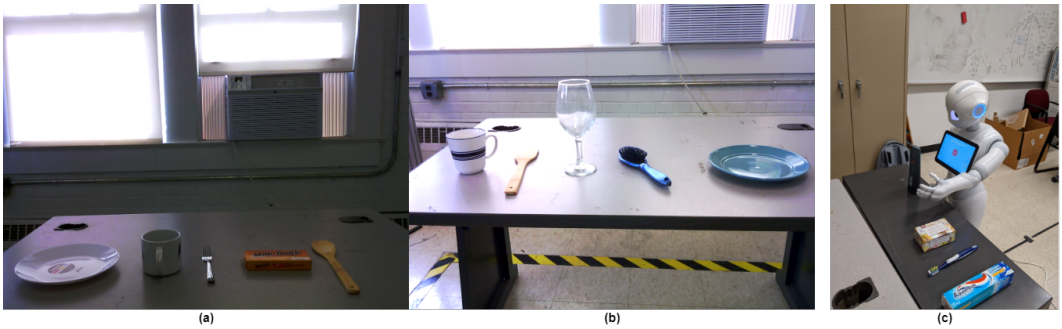

Figure 2: (a, b) Examples of different sets of objects present on a table in different lighting conditions. Note the variations in camera angle, background and arrangements of objects in the two images. (c) Pepper robot capturing different views of an object of class shampoo using its hand camera.

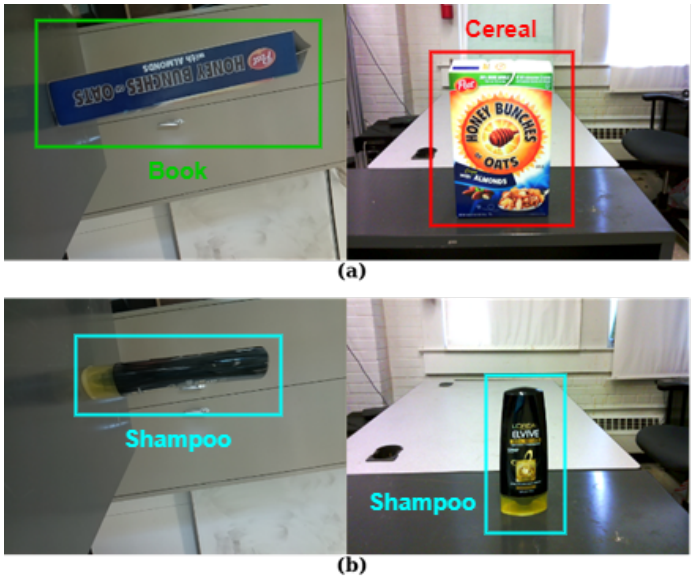

Figure 3: **Viewpoint Consistency:** Examples of multiple views of object classes (a) cereal and (b) shampoo. (a) Inconsistent predictions for different views of the same object generates a higher reward for acquiring the label of the object, while (b) consistent predictions generate a lower reward.

4.2 Manipulation and Image Capturing Module

As shown by Lomonaco et al. [1], multiple continuous views of different objects help a model develop a better representation which improves the recognition performance of the model. In the work proposed by Lomonaco et al. [1], a human holds an object in front of a camera and moves the object in his/her hand to get multiple views of the object. However, in real-world scenarios, an autonomous robot does not have access to unlimited human assistance. Thus, a robot must capture different views of an object autonomously. In this paper, we use the humanoid robot Pepper for the experiments. The head cameras of the Pepper robot, however, are not sufficient to capture the side views of the objects on a table. Hence, we attached a Raspberry Pi V2 camera to one of the robot's hands, so that the robot could move its hand around the objects to capture different views.

To allow the robot to capture different views of each object autonomously, we use the locations of the objects (from the object localization module) to allow the robot to move itself in front of the objects. We then allow the robot to move its arm around each object and capture the side views of the object using the camera on its hand. The objects were placed at the same height as Pepper's arms, so that same arm motion was used to capture images for all objects. We also use the head camera to capture the front/top views of the object. Figure 2 in the main paper shows examples of the images captured by Pepper's head camera and the hand camera. The arms were controlled using inverse kinematics

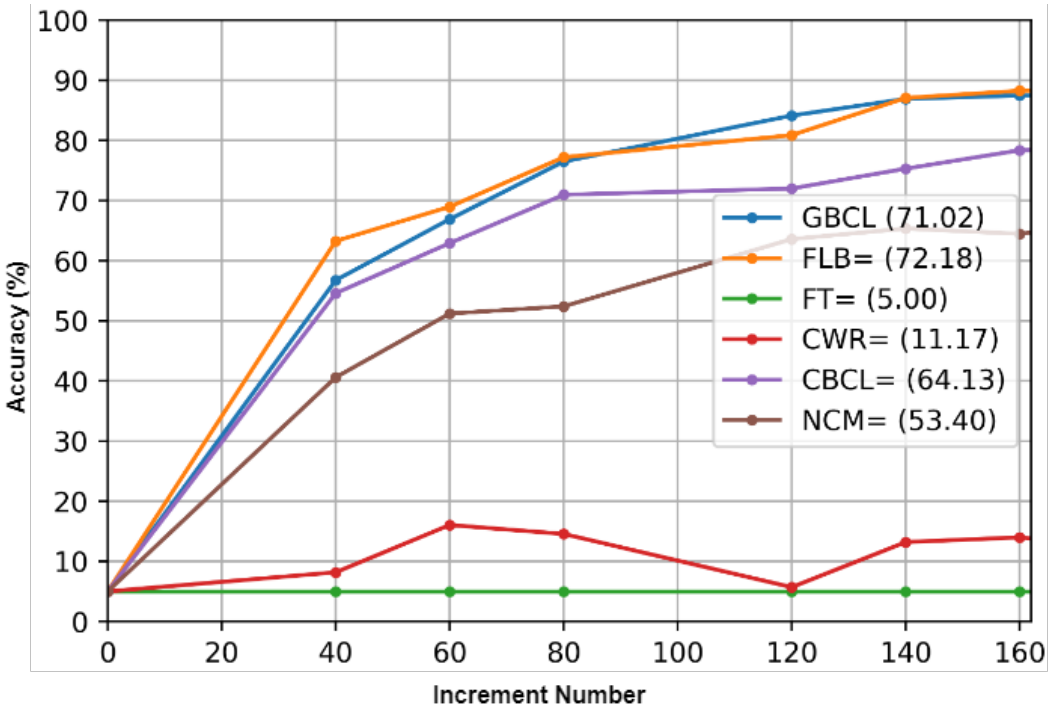

Figure 4: Comparison of our method (blue curve) to SOTA approaches in terms of classification accuracy on the Pepper dataset. Average incremental accuracy is reported in parenthesis.

module on the robot, where the robot moved the arms in a pre-defined motion in front of each object. Note that the robot could use its hand to grasp and move the object and capture images using its head camera. However, grasping and manipulating the object is a challenging problem and was not pursued as a part of this work.

After capturing multiple views of individual objects, the robot uses the AL module of our approach to find the most informative objects. The robot then asks the human user (experimenter) to provide the label of the object using the text-to-speech API. The user then provides the labels of the most informative objects as text input using a keyboard. Our system then updates the GMM representations and trains the classifier model on the newly labeled data. Details about our framework are in the main paper.

4.3 FoCAL Dataset from the Pepper Robot

As a part of the paper, we provide the complete dataset of object images captured by the Pepper robot (available here: `https://tinyurl.com/2vuwv8ye`). The complete dataset consists of 16988 images for 300 different objects belonging to 20 classes. We use 240 objects (13827 images) in the training set and 60 objects (3161 images) in the test set.

5 Experiment on the Pepper Dataset

We performed further experiments on the dataset collected by the Pepper robot, and compared GBCL with other approaches. For this experiment, we only report the accuracy for 160 increments, as the final accuracy for all the models starts to saturate. Figure 4 shows the comparison of GBCL with other approaches on the Pepper dataset. Similar to the results on CORe-50 dataset, GBCL is the closest to the batch learning approach (FLB) in terms of the final accuracy (less than 1% lower after 160 increments). CBCL is the next best approach, followed by NCM. As expected, FT and CWR suffer from catastrophic forgetting. These results again confirm the effectiveness of our approach for FoCAL.

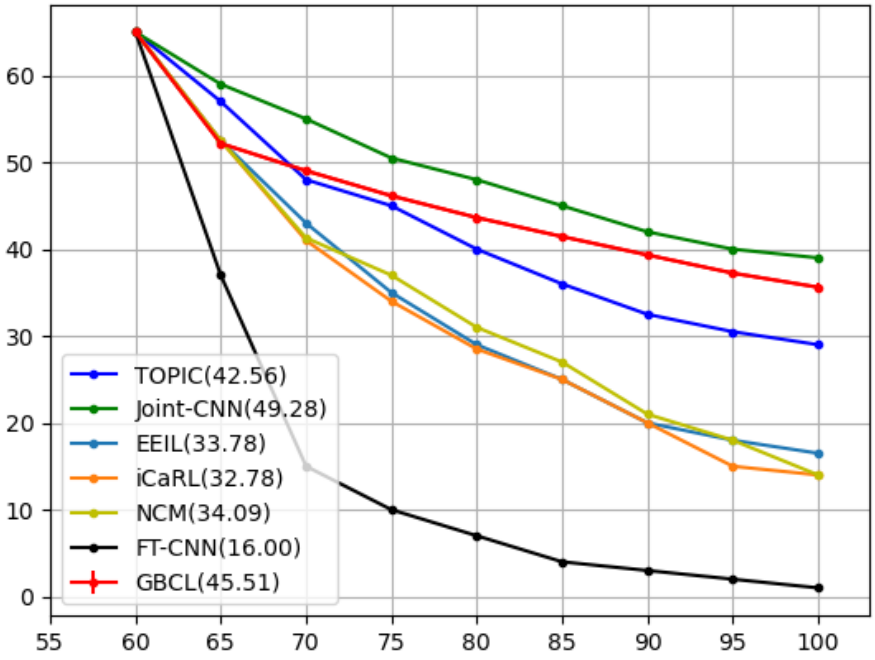

Figure 5: Comparison of GBCL (red curve) to other approaches in terms of classification accuracy on the CIFAR-100 dataset. Average incremental accuracy is reported in parenthesis. GBCL's curve shows average and standard deviation of 3 runs with random seeds.

6 Few-Shot Class Incremental Learning (FSCIL) Experiment

One of the limitations of our GBCL approach is that it uses a pre-trained feature extractor. However, there are some approaches for few-shot class incremental learning (FSCIL) that also continue to update the CNN representation from only a few examples per class. An argument can be made that such approaches might be better suited for FoCAL than GBCL as they also continue to adapt the feature representation. To further explore this idea, we performed a separate experiment with GBCL to compare against the FSCIL approaches. As the FSCIL approaches are not designed for FoCAL, for a fair compairson we tested GBCL for the standard FSCIL setup [12] on the CIFAR-100 dataset [3].

In the FSCIL setup [12], we first train a ResNet-18 from scratch on 60 out of 100 classes of the CIFAR-100 dataset. After training on the base classes, we freeze this network and use it as a frozen feture extractor in the next increments. After the base class training, the rest of the 40 classes are learned in 8 increments with 5 classes per increment. In the spirit of FSCIL, the 60 base classes are trained with 500 images per class, while the rest of the 40 classes are trained only with 5 images per class. Further, as all the images are labeled in each increment, we remove the active learning phase from GBCL and only use the GMM clustering and pseudo-rehearsal phases for this experiment. The training settings and the probability threshold hyperparameter were kept the same as in FoCAL experiments.

Figure 5 shows a comparison of GBCL (red curve) with batch learning upperbound (green curve), TOPIC [12] (blue curve) and other approaches on the CIFAR-100 dataset. Results for all the other approaches are reported from [12]. In the first increment, all the approaches achieve similar accuracy as they all train a ResNet-18 for 60 classes in the CIFAR-100 dataset. In the second increment, GBCL's accuracy drops significantly in comparison with TOPIC, because TOPIC updates the feature representation and adapts it to the classes learned in the second increment. In contrast, GBCL uses the CNN as a fixed feature extractor and therefore its representation is not fully adapted to the classes in the second increment. In the next increment, however, GBCL performs slightly better than TOPIC

because the representation learned by TOPIC is not general anymore and it became too specific to the small number of classes learned in the previous increment. In contrast, GBCL can still use the more general features learned in the first increment from a large number of classes. Further, the GMM clustering and the pseudo-rehearsal phase allow GBCL to continue to learn the complex distribution of the new classes without significant forgetting. In the subsequent increments, the accuracy gap between GBCL and TOPIC continues to increase, and after learning 100 classes GBCL achieves $\sim$8% higher accuracy than TOPIC. Another interesting result is that GBCL also closes the accuracy gap from the Joint-CNN (batch learning) approach over 8 increments. This shows that over a large number of increments GBCL can achieve similar performance to batch learning by continuing to learn the complex distributions of the object classes. These results show that adapting the feature representation using a few examples per class can be useful when learning for a single increment only. For a large number of increments, GBCL can achieve significantly better performance by using a fixed feature extractor with GMM representations and pseudo-rehearsal.

References

[1] Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78, pages 17–26, 2017.

[2] William H. Beluch, Tim Genewein, Andreas Nürnberger, and Jan M. Köhler. The power of ensembles for active learning in image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[3] Alex Krizhevsky. Learning multiple layers of features from tiny images, 2009. Technical report, University of Toronto.

[4] Ali Ayub and Alan R. Wagner. Cognitively-inspired model for incremental learning using a few examples. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[5] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[6] Z. Li and D. Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, Dec 2018.

[7] James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences of the United States of America*, 114(13):3521–3526, 2017.

[8] M. Dehghan, Z. Zhang, M. Siam, J. Jin, L. Petrich, and M. Jagersand. Online object and task learning via human robot interaction. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2132–2138, May 2019.

[9] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019.

[10] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[11] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[12] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.