Supplementary Materials - Incremental Deep Feature Modeling for Continual Novelty Detection (incDFM)

Amanda Rios[1,2], Nilesh Ahuja[2], Ibrahima Ndiour[2], Utku Genc[2], Laurent Itti[1], and Omesh Tickoo[2]

[1] University of Southern California, Los Angeles, CA 90089, USA
[2] Intel Labs, USA

1 Continual Novelty Detection Additional Results

1.1 Intra-dataset Novelty Detection Results

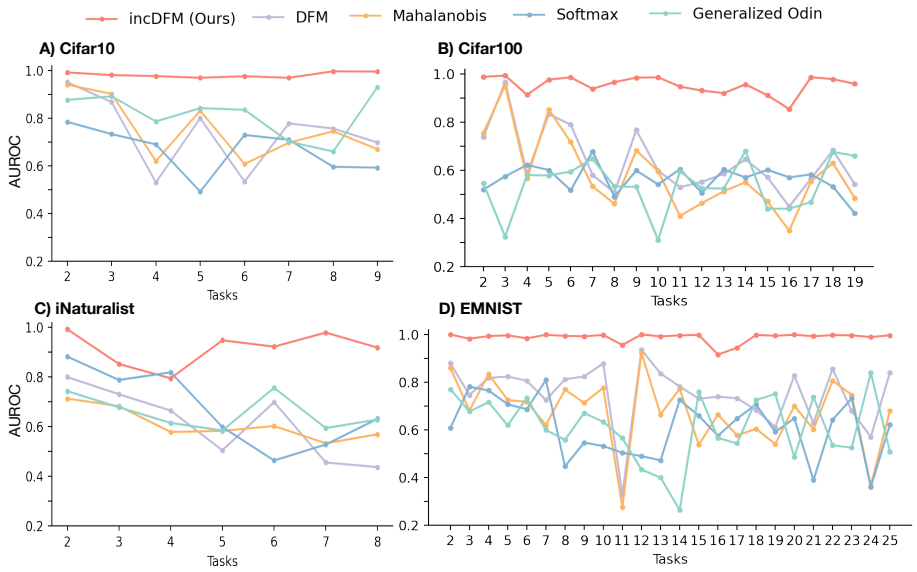

Fig. 1: Intra-dataset Novelty Detection - AUROC scores per task using test set. The test set is equivalent, in proportions (ratio old:new), to unlabeled train data used to fit to detected old samples. Also, in the case of incDFM, this is evaluated after all iterations are performed on the unlabeled train data.

We compute AUROC or AUPR scores using the test sets for each subset of the data corresponding to ID and OOD samples, only for sake of evaluation, $D_t^{test} = OOD_t^{test} \cup ID_t^{test}$. These are the most unbiased scores since the OOD

detector will not have been exposed to these samples ever before in previous tasks as holdout ID data (ID_t). Also, for fairness, in the case of incDFM we compute the test set AUROC and AUPR scores after completing all iterations for novelty estimation with the train set. That is, we use the test set only for testing after incDFM has completed all of its training and do not use it to compute incDFM intermediate parameters T_i^{new}. We reported the average over all tasks for the AUROC and AUPR test scores in the main paper Fig 4b table. All other experiments in the main paper report results using the test sets. Here, in Figure 1, we show the full per task result curves using test evaluation data and can observe that incDFM over-performs baselines through all tasks. The trend observed using test sets and train sets is similar.

1.2 Inter-dataset Novelty Detection Results

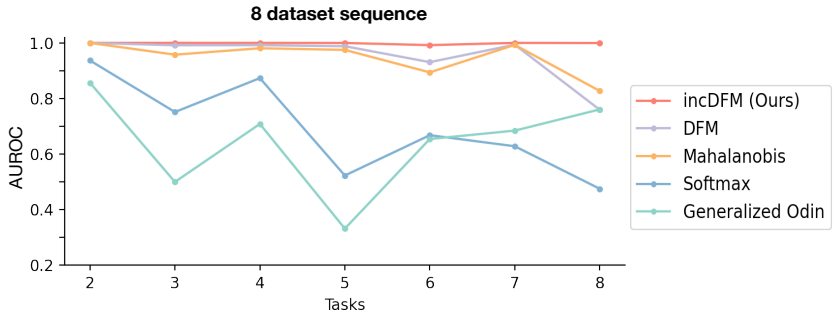

Fig. 2: Inter-dataset Novelty Detection (8 dataset sequence) - AUROC scores per task using test set equivalent, in proportions (ratio old:new), to unlabeled train data. In the case of incDFM, this is evaluated after all iterations are performed on the unlabeled train data.

In figure 2 we show AUROC scores per task for the 8 dataset sequence using test evaluation data. For Odin and Softmax baselines, we report results for the task-independent implementation (see section 4.1 of our main paper).

2 Estimating the stopping point for incremental novelty recruitment in incDFM

To estimate a stopping point to incremental recruitment, we use a validation set that contains only in-distribution (ID/old) samples and is updated by the algorithm at every task t, $u_t^{val} = \mathcal{F}(\{\mathbf{V_k^{val}}\}, k < t)$ where $\mathcal{F}$ is the feature extractor. In practice, at each task we reserve a small percentage of detected novel samples for validation and do not use them for fitting any parameters. The validation set is used to estimate, at each iteration, the total number of

P_{val}	95	85	75
cifar10	**94.6**	94.0	93.4
cifar100	87.4	**91.2**	90.0
emnist	**95.6**	95.2	94.3
iNaturalist	89.4	**89.7**	88.8

Table 1: incDFM F1 Scores - averaged across all tasks in intra-dataset class incremental experiments - when varying P_{val}

OOD samples left in the unlabeled pool, $N_{i,left}^{new}$ by a principle of exclusion, i.e, we set high validation threshold P_{val} on the high percentile range and estimate:

$$N_{i,left}^{new} = Count(S_i > Percentile(S_i^{val}, P^{val})) \tag{1}$$

$$indices_i^{new} = \text{argsort}_i(S_i)[:R] \quad \text{if} \quad N_{i,left}^{new} > 0 \tag{2}$$

Incremental recruitment cannot exceed N_i^{new} at each iteration. R is the recruitment percent per iteration. S_i are the composite scores for the unlabeled train data and S_i^{val} are the composite scores for validation data. Note that both S_i and S_i^{val} are computed equivalently using consolidated $\{\mathcal{T}_k, k < 1\}$ and incDFM's previous iterations new task parameters $\mathcal{T}_{t,i-1}$. For S_i^{val} this means:

$$S_{i,old}^{val} = \min_k FRE(\mathbf{u_t^{val}}, \mathcal{T}_k), k < t \tag{3}$$

$$S_{i,new}^{val} = FRE(\mathbf{u_t^{val}}, \mathcal{T}_{t,i-1}) \tag{4}$$

$$S_i^{val} = \frac{S_{i,old}^{val}}{\lambda S_{i,new}^{val}} \tag{5}$$

$$\tag{6}$$

Thus, validation scores also are affected by incremental estimation of $\widehat{OOD_t}$ since its samples, which are all ID, will tend to have increasingly higher $S_{i-1}^{new,val}$ values as the estimation of $\widehat{OOD_t}$ improves.

We show a hyperparameter sweep over a few percentile values from the validation set in Table 1. Overall, setting a high percentile value (ex: 95th or 85th percentile) tended to yield best results across datasets even though the difference between F1 scores in the 95-75 percentile range was subtle. Good results with high percentile values aligns with the assumption that OOD and ID data tend to separate over the course of iterations: at high val-ID thresholds, you still obtain a high precision and recall value for novel (OOD) data.

3 Thresholding in Baselines - Hyperparameter Sweep

For all four baselines, we select $\widehat{OOD_t}$ for task t by applying a single threshold on the corresponding generated uncertainty scores ($scores_i$), as originally intended

P_{val}	95	75	55	35
DFM	40.5	71.0	**84.5**	79.6
Mahal	42.8	67.7	**74.8**	73.8
Softmax	23.1	56.1	**70.1**	68.1
Odin	27.7	59.4	**66.4**	62.0

(a)

P_{val}	95	75	55	35
DFM	13.6	54.2	72.8	**81.7**
Mahal	24.7	49.3	60.4	**65.9**
Softmax	11.6	43.5	**64.4**	54.7
Odin	11.4	31.7	**56.7**	50.6

(b)

P_{val}	95	75	55	35
DFM	16.9	70.8	84.8	**88.4**
Mahal	27.0	57.8	66.8	**70.0**
Softmax	14.3	49.3	57.8	**61.9**
Odin	18.8	48.1	64.7	**66.2**

(c)

P_{val}	95	75	55	35
DFM	22.0	57.6	**73.71**	70.0
Mahal	20.4	47.8	**66.3**	59.4
Softmax	36.3	**68.2**	66.3	65.0
Odin	28.7	68.4	**71.77**	67.7

(d)

Fig. 3: Average F1 scores for baselines across tasks, when varying the validation threshold used during $\widehat{OOD_t}$ estimate - **(a) Cifar10, (b) Cifar100, (c) emnist, (d) iNaturalist** - The threshold is set as a percentile P_{val} of the validation set, the latter containing only ID data.

in the original implementation of these baselines. In our case, the threshold is chosen based on a validation set containing only in-distribution samples. The threshold is chosen to be equivalent to a certain percentile value of the validation set, P_{val}. As such, $\widehat{OOD_t}$ is estimated by:

$$Scores_i, Scores_i^{val} = \text{OODMethod}(\mathbf{u_t}, \mathbf{u_t^{val}}) \tag{7}$$

$$\widehat{OOD_t} = indices^{new} = \{i | Scores_i > Percentile(Scores_i^{val}, P^{val})\} \tag{8}$$

For fairness, we employ the same validation set $u_t^{val} = \mathcal{F}(\{\mathbf{V_k^{val}}\}, k < t)$ used by incDFM (Main paper section 3.1.3, Supplementary section 2), where $\mathcal{F}$ is the feature extractor. For all baselines we perform a hyperparameter sweep over thresholds, results are shown in Figure 3. In the main paper we report best results for each baseline. We show that the single-threshold baseline novelty detectors, in general, tend to perform better with a low threshold. This is likely because ID_t and OOD_t scores are very enmeshed and a high threshold will result in very low recall value, insufficient for novelty characterization going forward.

4 Feature Extraction Network

We experimented with different feature extraction networks. Overall, incDFM and baselines on average performed best with a frozen Resnet50 backbone pretrained using contrastive learning [SWAV - [2]] in comparison to fine-tuning, for both continual and offline experiments. Table 2 compares amongst feature extraction approaches (plastic/finetune vs frozen) for offline OOD detection (see main paper section 5.1).

The results on Table 2 are aligned with recent advances in the transfer/adaptive learning literature suggesting that most features needed for natural-image datasets

Cifar10 → SVHN	incDFM	Mahal	Softmax
Frozen-Resnet-50-SWAV	**99.9**	93.1	88.2
Finetune-Resnet-50-SWAV	99.3	95.03	71.4
Frozen-Resnet-50	99.8	60.0	87.3
Finetune-Resnet-50	99.9	87.7	76.0

Table 2: AUROC scores for offline OOD estimation

can be found in rich pre-trained-on-imagenet backbones [4]. Moreover, in a task-independent CL setting, using a coreset to estimate past data can lead to overfitting. Thus, freezing the backbone has become a common practice in the CL literature [12, 14]. For OOD methods relying on classification (Odin, Softmax), we use a plastic/trainable 1-hidden-layer MLP (hidden dimension of 4096 units) to learn the class mapping. Similarly for the end-to-end unsupervised class-incremental classification pipeline.

5 End-to-end Unsupervised class-incremental classification Pipeline

5.1 Memory Coreset

We employ a similar memory coreset building scheme as in [11]. We keep a small memory coreset with embeddings and pseudolabels $\mathcal{C}+1$ corresponding to past tasks' $\widehat{OOD}_k, k < t$ (novelty detections - see main paper section 3.2). At each task, a selection method is employed to choose which samples detected as novel will go into the coreset, with the aim to maximize sample heterogeneity since the coreset has a fixed size. We use K-means clustering per pseudolabel to select samples for storage and for removal. At each task, when $\widehat{OOD}_t$ is detected, we run k-means clustering, super-labeling the embeddings of $\widehat{OOD}_t$ as one of K clusters. At the time of insertion into the memory coreset, we select equal numbers of samples from each cluster. Additionally, if the coreset is full we compute the space needed for new samples and remove an equivalent number of old embeddings. We do this by assessing their stored super-cluster labels and removing equal amounts of samples per novelty pseudolabel and per cluster, thereby preserving heterogeneity. By storing the per-novelty, cluster assignment superlabels we also avoid repeating the clustering operation.

5.2 Experience replay

At each task, our model is trained using the current tasks's predicted $\widehat{OOD}_t$ and select memory embeddings of past tasks' $\widehat{OOD}_k, k < t$ present in the memory coreset. This forms an extended training set S_t that is used to minimize the cross-entropy loss for classification (equations 9,10). Note that we use the novelty pseudolabels as targets for the classifier.

$$S_t = OOD_t \cup \lambda_{mem} OOD_{t-1}^{memory} \tag{9}$$

$$\theta_t^* = \min_{\theta_t} L(\theta, S_t) \tag{10}$$

The memory component can be given a weighted importance, λ_{mem}. We typically set λ_{mem} to reflect the proportion of classes present in the coreset.

Observation - OOD baselines that rely on classification: We employ the same procedure described in 5.1. and 5.2. to train Odin and Softmax novelty detectors in intra-dataset class incremental experiments.

6 Inter-Dataset novelty detection using the 8 datasets - Experimental Clarification

In this experiment, we wanted to analyze the ability of incDFM and baselines to detect novelty continually in the setting where each novelty is an entire new dataset. This proposed CL scenario is closer to the traditional offline OOD/ID detection, which typically also consider an entire novel dataset as OOD data. In the main paper we compare this inter-dataset experiment with our intra-dataset continual learning. For the inter-dataset experiment we consider a sequence of eight tasks each being one of 8 object recognition datasets as in [1]. Each of the 8 datasets in the order presented: **1.** Oxford *Flowers* [9] for fine-grained flower classification with 102 classes; **2.** MIT *Scenes* [10] for indoor scene classification with 67 classes; **3.** Caltech-UCSD *Birds* [13] for fine-grained bird classification with 200 classes; **4.** Stanford *Cars* [6] for fine-grained car classification with 196 classes; **5.** FGVC-*Aircraft* [7] for fined-grained aircraft classification with 70 classes; **6.** VOC *actions* [5], the human action classification subset of the VOC challenge 2012 with 10 classes; **7.** *Letters*, the Chars74K datasets [3] for character recognition in natural images with 62 classes; and **8.** the Google Street View House Number *SVHN* dataset [8] with 10 classes. The total 8 dataset sequence contains a total of 227,597 pictures in 717 classes.

References

1. Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., Tuytelaars, T.: Memory aware synapses: Learning what (not) to forget. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 139–154 (2018)
2. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. Advances in Neural Information Processing Systems **33**, 9912–9924 (2020)
3. De Campos, T.E., Babu, B.R., Varma, M., et al.: Character recognition in natural images. VISAPP (2) **7** (2009)
4. Evci, U., Dumoulin, V., Larochelle, H., Mozer, M.C.: Head2toe: Utilizing intermediate representations for better transfer learning. arXiv preprint arXiv:2201.03529 (2022)

5. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. International journal of computer vision **111**(1), 98–136 (2015)
6. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 554–561 (2013)
7. Maji, S., Rahtu, E., Kannala, J., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. arXiv preprint arXiv:1306.5151 (2013)
8. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning (2011)
9. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing. pp. 722–729. IEEE (2008)
10. Quattoni, A., Torralba, A.: Recognizing indoor scenes. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 413–420. IEEE (2009)
11. Rios, A., Itti, L.: Closed-loop memory gan for continual learning. arXiv preprint arXiv:1811.01146 (2018)
12. Rios, A., Itti, L.: Lifelong learning without a task oracle. In: 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI). pp. 255–263. IEEE (2020)
13. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011)
14. Wen, S., Rios, A., Ge, Y., Itti, L.: Beneficial perturbation network for designing general adaptive artificial intelligence systems. IEEE Transactions on Neural Networks and Learning Systems (2021)