incDFM: Incremental Deep Feature Modeling for Continual Novelty Detection

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

Abstract. Novelty detection is a key capability for practical machine learning in the real world, where models operate in non-stationary conditions and are repeatedly exposed to new, unseen data. Yet, most current novelty detection approaches have been developed exclusively for static, offline use. They scale poorly under more realistic, continual learning regimes in which data distribution shifts occur. To address this critical gap, this paper proposes incDFM (incremental Deep Feature Modeling), a self-supervised continual novelty detector. The method builds a statistical model over the space of intermediate features produced by a deep network, and utilizes feature reconstruction errors as uncertainty scores to guide the detection of novel samples. Most importantly, incDFM estimates the statistical model incrementally (via several iterations within a task), instead of a single-shot. Each time it selects only the most confident novel samples which will then guide subsequent recruitment incrementally. For a certain task where the ML model encounters a mixture of old and novel data, the detector flags novel samples to incorporate them to old knowledge. Then the detector is updated with the flagged novel samples, in preparation for a next task. To quantify and benchmark performance, we adapted multiple datasets for continual learning: CIFAR-10, CIFAR-100, SVHN, iNaturalist, and the 8-dataset. Our experiments show that incDFM achieves state of the art continual novelty detection performance. Furthermore, when examined in the greater context of continual learning for classification, our method is successful in minimizing catastrophic forgetting and error propagation.

Keywords: Continual Learning; Out-of-distribution detection

1 Introduction

Deep Neural network models excel at learning complex mappings between inputs and outputs, so long as the data is drawn from a stationary distribution. Yet, when these models are deployed in the real-world, they may encounter out-of-distribution (OOD, "novel") inputs, i.e. input data that does not resemble the training data (in-distribution, "ID"), prompting misleading predictions. This is a strong limitation because many real world applications require handling non-stationary data. Models deployed in self-driving cars, for instance,

will inevitably encounter novel out-of-distribution data (e.g. new terrains, objects, weather) that they have to adapt to. Hence, continual novelty detection is critical for operating in real-world, non-stationary conditions. However, most novelty detection methods were developed for and evaluated against a single fixed split of ID/OOD data. They do not integrate the detected OOD data into the learnt knowledge and perform poorly in dynamic, non-stationary conditions. On the other hand, most approaches in continual learning (CL) focus on mitigating *catastrophic forgetting*, a phenomenon in which training a neural network on a new task with novel data typically destroys the fixed mapping learned from the previous tasks. Most importantly, they use an oracle to identify novel data, leaving the question of continual novelty detection largely unaddressed.

We seek to bridge the divide between the continual learning and novelty detection fields by addressing novelty detection in continual learning, a much more challenging evaluation and deployment paradigm. Specifically, we focus on the task-incremental continual learning setting, where the model increasingly encounters new, additional classes of data without significant distribution shift for the already-seen classes of previous tasks. In this setting, a novelty detector is presented with several OOD/ID separation tasks through time. This can bring about several challenges: (1) *Novelty consolidation*: integrating detected novel samples to knowledge (to avoid treating them as novel in subsequent tasks) (2) *Catastrophic forgetting*: remembering this cumulative knowledge through tasks, and (3) *Error propagation*: minimizing the number of samples falsely flagged as novel to avoid impairing knowledge consolidation.

Contribution: We propose a novelty detection algorithm, "incremental Deep Feature Modeling" (incDFM) that addresses these three challenges. It is trained using only ID data and designed to operate under the continual learning setting. incDFM builds a per-class or per-task statistical model over the space of intermediate features produced by the deep network and computes a feature reconstruction score to flag the OOD samples. Most importantly, with the goal of minimizing continual error propagation, incDFM estimates this statistical model incrementally (via several iterations within a task) for each novel task. At each iteration within a novel task, it recruits the top most "certain" novel samples that will then improve subsequent recruitments incrementally. incDFM can be used to substitute the novelty oracle used in traditional supervised CL. Finally, we show that incDFM achieves state of the art novelty detection performance when evaluated on multiple datasets adapted for task-based continual learning, such as CIFAR-10, CIFAR-100, SVHN, iNaturalist and the 8-dataset.

2 Background and Motivation

Novelty Detection: Also known as outlier or out-of-distribution (OOD) detection, novelty detection is a very active research area. It is typically performed by making the network provide an uncertainty score (along with the output) for each input. Common methods include the Softmax score [15] and its temperature-scaled variants such as ODIN [24]. Bayesian neural networks [12] and ensembles

of discriminative classifiers [22] can generate high quality uncertainty, but at the cost of complex model representations, and substantial compute and memory. Deep generative models learn distributions over the input data, and then evaluate the likelihood of new inputs with respect to the learnt distributions [16], [32], [37]. Gradient-based characterization of abnormality in autoencoders is highlighted in [21]. Finally, there are methods [1], [23] that learn parametric class-conditional probability distributions over the features and use the likelihoods (w.r.t the learnt distributions) as uncertainty scores.

Continual Learning: This paper focuses on *task incremental learning*, a paradigm where a model continually learns from a sequence of tasks that each introduce novel data but with no or limited access to past, labeled data. The majority of the CL literature has focused on *catastrophic forgetting* [11, 33] while mostly offloading the task transition detection duty to a so-called *novelty oracle*. Overall, [39] proposes that current continual learning algorithms can be grouped into *task-dependent* and *task-independent* models by their reliance on task labels at test time. *Task-independent* algorithms do not require task labels and typically employ a single shared classification layer which has as many output nodes as the number of learned classes over all tasks. One subclass consists of *regularization-based* approaches which aim to mitigate forgetting by constraining the change of learnable parameters. Alternatively, *replay-based* algorithms approximate the CL problem to a multi-task setting by either storing [25, 36] or learning to generate [41, 46, 38] past data. Broadly, task-independent models solve a more challenging CL formulation since task-specific parameters are not exploited for test time performance. *Task-dependent* methods, on the other hand, require the availability of task labels which are usually provided by a *task oracle* and utilize this information by employing task-specific classification heads and other task-dependent parameters to share the rest of the network for different tasks, e.g. partitioning with context [48, 47, 5] or mask matrices [27, 9]. Dependence on an oracle limits their applicability as determining tasks and detecting task transitions are challenging and also prone to *forgetting* [39].

Continual Novelty Detection: The problem of novelty detection in the continual learning setting has not been extensively studied or discussed in the literature. Most CL literature has assumed the use of a novelty oracle to indicate fully-labeled task transitions. Incipient proposals and discussions for novelty detection have occurred in [42, 3, 28]. Yet, most of these works do not propose novel OOD algorithms, rather, they adapt existing OOD approaches and to a limited success. For instance, the closest work, by [3], compares among several existing OOD detectors. However, their best results occur under a task-oracle-dependent continual learning setting and using task-dependent CL algorithms to mitigate novelty detection forgetting. We argue that this limits real-world applicability since it is not realistic to assume that at deployment, unlabeled test samples will be accompanied by their respective task IDs, which would obviate the need for OOD detection in the first place. To our knowledge, no work has yet proposed a reliable OOD detection solution for oracle-less continual learning over several tasks, a more realistic but also more challenging setting.

3 Methodology for Continual Novelty Detection

To bridge the gap between the CL and OOD fields, we first establish a novelty detection methodology suited to continual learning. In our framework, each incoming task t is an unsupervised mixture of unseen ID_t samples ("old" classes) and OOD_t samples ("new" unseen classes):

$$ID_t = \{u^{old,unseen} | u^{old,unseen} \sim D_k\}, k = 1, ..., t-1$$
$$OOD_t = \{u^{new,unseen} | u^{new,unseen} \sim D_t\}$$

ID_t comprises unseen samples that were never used in training, but come from the same source distributions $D_k, k = 1, \ldots, t-1$ that were used to train past tasks, while OOD_t consists of samples from an entirely new distribution D_t. In our experiments, we simulate this by using 80% of original training samples as novel data at each task and leaving the remainder for introduction at later tasks (at which point they will be old ID data). The goal for the novelty detector is to accurately differentiate between ID_t and OOD_t to produce an estimate of the novel samples which we denote as $\widehat{OOD}_t$. This then becomes the training data to consolidate knowledge of novel samples.

This methodology leads to the additional challenges of *catastrophic forgetting* and *error propagation* (alluded to in Section 1) that aren't present in conventional offline OOD detection. First, as more and more classes/tasks are encountered, incDFM has to increasingly add to its stored representation of what is ID and remember the cumulative $\{D_k\}, k \leq t$ going forward. If past D_k's are not properly remembered and represented in knowledge, this can result in catastrophic forgetting and failure to identify incoming old samples as ID. incDFM addresses this by building a per-class or per-task statistical model to detect novel samples at each task. The per-task parameters once stored are not interfered with in future tasks, minimizing forgetting (refer to section 3.1.2). Second, as already mentioned, whenever the novelty detector finalizes its selection of novel samples $\widehat{OOD}_t$, these are then used as training data to consolidate knowledge and expand what is considered as ID_{t+1} for the following task. However, since $\widehat{OOD}_t$ can contain misclassified samples, this could result in an inaccurate representation of D_t during consolidation, which will lead to error propagation that grows progressively worse. Cumulatively, these two aspects can lead to severe performance degradation. We show incDFM's incremental recruitment strategy (section 3.1.2) minimizes error propagation.

Lastly, we also evaluate continual OOD detection in an inherently more difficult experimental paradigm where ID and OOD sets are drawn from different splits of the same dataset (intra-dataset). In particular, we propose experiments of intra-dataset class-incremental learning where, at each task, only one novel class is introduced, up until all classes of a dataset are covered. ID and OOD splits sampled from the same dataset tend to be close and harder to disentangle [39]. In contrast, most offline OOD detection literature has focused on OOD/ID splits between different datasets (inter-dataset, e.g., CIFAR-10 as ID vs. SVHN as OOD) - these are typically comprised of highly divergent data distributions,

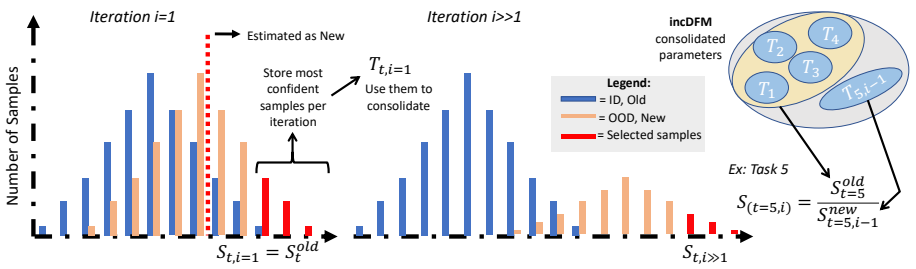


Fig. 1: incDFM estimates novelty incrementally per task. A tasks's unlabeled data mixture is shown here with ID/old samples in blue and OOD/novel samples in orange. At each iteration within one novel task, incDFM recruits the top most "certain" novel samples (in red) according to the evaluation function S_i. It then removes them from the unlabeled pool. At iteration 1 we can see new and old distributions are entangled but tend to separate in later tasks, as incDFM improves its estimate of novelty.

causing the model to first explore accidental low-level statistical differences instead of more meaningful semantic variances. Overall, combining a naturally harder ID/OOD setting per task with having to remember what is ID through time makes most conventional OOD detectors underperfom. In incDFM, iterative estimation and recruitment algorithm is better suited to continual and challenging ID/OOD splits.

3.1 incDFM Model

3.1.1. Deep feature Modeling incDFM is built upon the OOD detection technique proposed in [1] based on probabilistic modeling of deep features. Consider a deep neural network (DNN) trained on an N-class classification problem. For an input $\mathbf{x}$, let $\mathbf{u} \triangleq \mathcal{F}_l(\mathbf{x})$ denote the output at an intermediate layer l of the network. In [1], class-conditional probability densities are learnt on this set of intermediate deep-features and the likelihood scores from these are used to discriminate between ID and OOD samples. A principal component analysis (PCA) transformation, $\mathcal{T} : \mathcal{H} \rightarrow \mathcal{L}$, is simultaneously learnt to map the high-dimensional features onto an appropriate lower-dimensional subspace, $dim(\mathcal{L}) \ll dim(\mathcal{H})$,prior to density estimation. The PCA transformations are also learnt on a per-class basis. For incDFM, this implies that a separate PCA transformation,$\mathcal{T}_t$, is learnt for each task t. In [29], it was shown that the *feature reconstruction error* (FRE) score, defined as

$$FRE(\mathbf{u}, \mathcal{T}) = \|\mathbf{u} - (\mathcal{T}^{\dagger} \circ \mathcal{T})\mathbf{u}\|_2 \tag{1}$$

is highly effective at discriminating between ID and OOD samples, where $\mathcal{T}^{\dagger}$ is the inverse PCA transformation (computed as the Moore-Penrose pseudo-inverse of $\mathcal{T}$) . The intuition behind FRE is that OOD samples will lie outside the subspace of ID samples and will hence result in higher FRE scores.

3.1.2. Knowledge consolidation and storage: To obtain deep features, incDFM employs a frozen feature extractor pre-trained via unsupervised contrastive learning on an independent large dataset - e.g., imagenet. Using a frozen pre-trained deep feature extractor showed superior performance to fine-tuning, which is in line with recent findings in the adaptive learning field [34, 8]. At each task t, we process all unlabeled samples, $\mathbf{x_t} = OOD_t \cup ID_t$ through the feature extractor and collect deep features $\mathbf{u_t} = \mathcal{F}_l(\mathbf{x_t})$ which are used as input to the main incDFM algorithm. Further, as mentioned earlier, we learn and store the parameters for $\mathcal{T}_t$ for each task separately (Procedure *Consolidate* in Algorithm 1 Fig 2). This consolidation approach has two advantages for continual learning. First, by modeling OOD_t via isolated per-task (per-class) parameters, we minimize catastrophic interference when new classes are introduced later on. The consolidated per-class parameters are never altered so cannot actually be "forgotten", assuming no distribution shift for old tasks. In deep neural networks (DNNs), by contrast, the majority, if not all, of parameters are shared between the classes and per-class importance of each weight is not as easily assessed. As such, when new classes are introduced, it is naturally much more difficult to isolate inter-class interference in DNN weight space. This is one of the reasons most CL approaches tackling single-headed classification require a replay strategy to not "forget", which can quickly escalate in memory usage. This brings us to the second advantage: our consolidation approach is both fast and memory-efficient. More specifically, it is fast because it requires a single PCA fitting operation per task. Additionally, it entails a low memory usage since it only retains the PCA transformation $\mathcal{T}_t$ per task, which is almost always less memory expensive than storing raw image samples for replay typical in task-independent CL approaches.

3.1.3. Novelty Detection and Selection: Incremental recruitment
When a new task arrives, the stored consolidation parameters ($\{\mathcal{T}_k\}$ for $k = 1 : t-1$) are used to initialize an incremental recruitment of novel samples. We express the unlabeled deep features from incoming task t as $\mathbf{u_t} = \mathcal{F}_l(\mathbf{x_t}), \mathbf{x_t} = ID_t \cup OOD_t$. For all unlabeled samples u_t, we first compute the FRE scores for all $k = t-1$ stored sets of transforms and then take the minimum FRE:

$$S^{old}(\mathbf{u_t}) = \min_k (FRE(\mathbf{u_t}, \mathcal{T}_k)), k = 1, ..., t-1 \tag{2}$$

Intuitively, this indicates which of the older classes/tasks each sample is closest to. We sort the set of unlabeled samples by their FRE scores with the intuition that ID_t samples will tend to yield lower values of FRE than OOD_t. We could presumably set a threshold and select samples whose scores exceed that to constitute $\widehat{OOD}_t$ and then estimate $\mathcal{T}_t$ from those. A relaxed threshold could result in $\widehat{OOD}_t$ containing a large number of ID samples misclassified as novel, whereas a high threshold might result in very few novel samples being available for computation of $\mathcal{T}_t$. Either way, this could lead to poor estimates of $\mathcal{T}_t$ and the error from this would propagate and progressively worsen for subsequent tasks.

Hence, we propose an iterative method to estimate the novel samples in an incremental fashion as outlined in Fig 2. In the first iteration, $i = 0$, we compute

Algorithm 1: incDFM - Incremental Novelty Recruitment per Task

Input : $\mathbf{u_t}$ - Deep Features of current Task
Require : I - Maximum Number of iterations; R - Recruitment per iteration;
Initialize : $S^{old} \leftarrow \text{KnowledgeScores}(X_t, \{\mathcal{T}_k\} \text{for } k < t)$;
$i \leftarrow 1$; $N^{new}_{1,left} \leftarrow length(X_t)$; $S_1 \leftarrow S^{old}$;
$\text{Indices}_t = [1, ..., length(X_t)]$; $\text{Indices}^{new}_1 = []$;

```
// Select most certain novel samples per iteration until stopping criterion
```

while $(i < I)$ *and* $(N^{new}_{i,left} > 0)$ **do**

```
    // Concatenate newly selected indices to previously selected
```

$\text{Indices}^{new}_i, N^{new}_{i,left} \leftarrow \text{SelectTop}(S_i, R)$

```
    // Remove selected indices from unlabeled pool
```

$\text{Indices}^{new} \leftarrow [\text{Indices}^{new}_i, \text{Indices}^{new}]$
$\text{Indices}_t \leftarrow \text{Indices}_t - \text{Indices}^{new}_i$
$\mathcal{T}_i \leftarrow \text{Consolidate}(\text{Indices}^{new}, X_t)$
$S^{new}_i \leftarrow FRE(\mathbf{u_t}, \mathcal{T}_i)$
$S_i \leftarrow \frac{S^{old}}{\lambda S^{new}_i}$
$i \leftarrow i + 1$

$\{\mathcal{T}_k\}, k = 1, .., t \leftarrow \text{Store}(\mathcal{T}_{t,I})$

Fig. 2: Procedures *KnowledgeScores* and *SelectTop* are described in section 3.1.3; *Consolidate* in 3.1.2

$S^{old}(\mathbf{u_t})$ as previously described in equation 2 (Procedure *KnowledgeScores*) and select only the highest R percent of the S_{old} scores, corresponding to the most confident "novel" samples until now (farthest from old). These samples constitute a first estimate, $\widehat{OOD}_{t,0}$, of what is OOD. They are used to consolidate knowledge by computing $\mathcal{T}_{t,0}$ and using the latter to obtain $S^{new}_0 \triangleq FRE(\mathbf{u_t}, \mathcal{T}_{t,0})$. For all subsequent iterations $i \geq 1$, we compute a composite evaluation score function S_i which combines S_{old} and the previous iteration's S^{new}_{i-1}

$$S_i = \frac{S^{old}}{\lambda S^{new}_{i-1}}, \quad S^{new}_{i-1} \triangleq FRE(\mathbf{u_t}, \mathcal{T}_{i-1}) \tag{3}$$

and use this composite score to select the next R top percent, $indices^{new}_i$ (Procedure *SelectTop* in Algorithm 1), which are then concatenated to all previous iteration's indices, $indices^{new}$, and used to compute the next estimate of $\mathcal{T}_{t,i}$. The idea behind this algorithm is to increasingly separate hard ID/OOD splits (Fig 1). At each iteration, OOD (novel) samples will tend to have low scores S^{new}_{i-1} and high S_{old}, resulting in the highest composite S_i values. To minimize errors, we set R conservatively to recruit only the most confident OOD detections. Moroever, as more and more confident OOD estimated samples are recruited, i.e. $Indices^{new}$ grows in size, the better will be the subsequent estimate of PCA parameters $\mathcal{T}_i$. This in turn will yield progressively more reliable S^{new}_i scores.

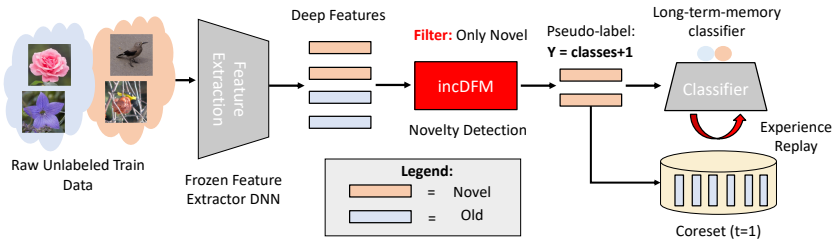


Fig. 3: Full Pipeline - unsupervised class incremental learning with incDFM

To estimate a stopping point to incremental recruitment, we set a total maximum number of iterations and employ a small validation set with only in-distribution (old) samples, $(\{V_k\}, k < t)$, to estimate if there is still a probability of having non-recruited novel samples left (suppl.). 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. For fairness, the same validation set is used across all baselines that we compare with.

3.2 Full Pipeline: unsupervised class-incremental learning using incDFM for continual novelty detection

We show that incDFM can be coupled onto an unsupervised class incremental classification pipeline, Figure 3. We take the same experimental setting previously described, where at each task we have a mixture of holdout samples of old classes and one new class at a time, all unlabeled. Over tasks, we keep a counter of how many novelties have been introduced so far, $\mathcal{C}_t$ (equivalent to number of classes in this case). At each task, after incDFM has selected a final estimate of novel samples $\widehat{OOD}_t$, these are pseudolabeled as $\mathcal{C}_{t-1} + 1$ and the counter is also incremented. As the classifier we use a perceptron on top of the frozen feature extractor that is also shared with incDFM, similarly to [39, 45]. The detected novel samples are then used to train the classifier using the pseudolabels as targets and are stored in a coreset for replay at future tasks. We employ a fixed size coreset with the same building strategy as in [38]. Thus, at each task, the classifier is trained using the current tasks detected $\widehat{OOD}_t$ samples and the samples in the coreset using experience replay to mitigate forgetting (suppl.).

4 Experiments

Intra-dataset class-incremental experiments: For intra-dataset experiments, we consider four datasets: 1. CIFAR-10 (10 classes), 2. CIFAR-100 (super-class level, 20 classes) [20], 3. EMNIST (26 classes) [6] and 4. iNaturalist21 (phylum level, 9 classes) [43]. We adapt all datasets for class-incremental learning by starting with 2 classes for the first task and adding one class at each incremental task until all classes are covered.

Inter-dataset Experiment: In this experiment, the novelty per task is an entire novel dataset (with multiple new classes). This is a CL version of the conventional ID/OOD setting. We compare how much easier it is to detect ID/OOD shifts in this CL inter-dataset paradigm versus the previous CL intra-dataset class incremental experiment. We consider a sequence of eight tasks each being one of 8 object recognition datasets (Flowers [31]; Scenes [35]; Birds [44]; Cars [19]; Aircrafts [26]; VOC Actions [10]; Letters [7]; svhn [30]) as in [2] (see suppl.).
Baselines: We compare and benchmark our method against the various commonly used offline OOD detectors: (i) Mahalanobis based OOD detector [23] (ii) Softmax based OOD detector [15] which uses softmax output as a confidence score, and (iii) Generalized ODIN [17] which introduces a decomposed softmax scoring function as an improvement of Softmax. Note that while Softmax and ODIN both rely on classification layers to detect novelties, Mahalanobis relies on distance scores computed from intermediate features of a DNN. For ODIN and Softmax we use the same classifier architecture as in our full-pipeline (section 5.3.), i.e. a perceptron (MLP) on top of the frozen feature extractor. Since these baselines were developed for offline OOD detection, we make the necessary adaptations to use them in continual learning: First, for Mahalanobis we keep a coreset with select past ID samples to estimate the joint covariance needed for the metric. Second, because the MLP classifier in both Softmax and in ODIN is plastic and updated continually, *catastrophic forgetting* is expected to have a degrading effect on continual novelty detection performance unless an alleviation mechanism is employed. In the intra-dataset class incremental experiments, we apply *coreset-based experience replay* [40], the same CL strategy as in our full-pipeline. Task-dependent algorithms cannot be applied in this case since each task is only one class. Alternatively, to mitigate forgetting in the inter-dataset experiment, we use PSP [5](a task-dependent CL algorithm) and separate readout heads per task, similar to [39]. The original PSP formulation requires a task oracle. Hence, we also propose a version of PSP that is oracle-less: we loop through all PSP task-conditioned MLP partitions and output heads collecting task-dependent Softmax/ODIN scores. We then select the task-dependent score yielding maximum certainty among them as a final task-independent score. Finally, we also compare against a direct implementation of DFM, which uses the same per-task knowledge consolidation strategy as described in Section 3.1.2. but does not employ our proposed incremental recruitment algorithm. This serves as an ablated version of incDFM. For all four baselines, we select $\widehat{OOD}_t$ by applying a single threshold per task on the corresponding generated uncertainty scores. The threshold is chosen based on a validation set containing ID samples. For fairness, we employ the same validation set $\{V_k\}, k < t$ used by incDFM. Refer to suppl. for more implementation details.
Architecture and Training Parameters: For all methods considered, including ours, we use a ResNet50 [14] backbone pre-trained on ImageNet using SwAV [4], a contrastive learning algorithm. For OOD methods which rely on classification (*ODIN*, *Softmax*) and also for the end-to-end class incremental learning pipeline, we use an MLP, with 4096-dimensional hidden layer, as the classifier.

The backbone is kept frozen for all tasks and only the classifier is fine-tuned over the course of an experiment. We compared to fine-tuning the backbone continually using experience replay but the reported frozen backbone approach worked best for incDFM and baselines (see suppl.), in line with the results reported in [13]. We optimize using ADAM [18] with a learning rate of 0.001 and decrease the learning rate when on a plateau. Finally, for all methods requiring a coreset, .e.g., our end-to-end incremental learning pipeline (section 5.3), ODIN, Softmax and Mahalanobis, we keep between 5-10% of the dataset converted to deep embeddings (output of frozen feature extractor) into a fixed-size coreset. For the end-to-end-pipeline, we train until convergence using 40 epochs per task for ODIN and 20 epochs per task for all others.

5 Results

5.1 Preliminary offline evaluation of incDFM and baselines

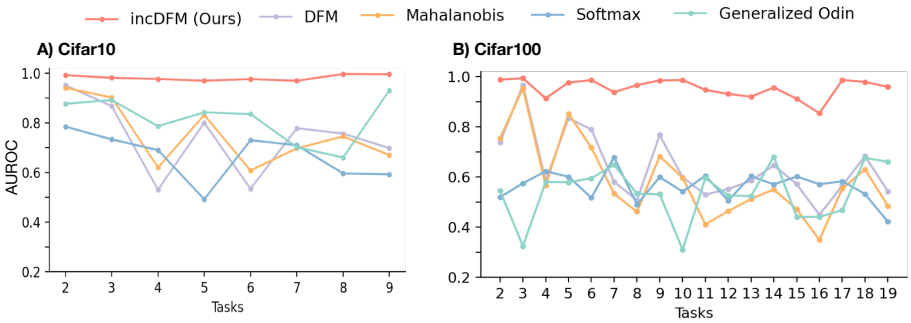
ID → OOD	incDFM	DFM	Mahal	Softmax	ODIN
CIFAR-10 → SVHN	**99.9**	93.4	93.1	88.2	95.8
CIFAR-100 → SVHN	**99.9**	93.6	87.7	83.5	88.4

Table 1: AUROC scores for offline OOD estimation

In table 1 we evaluate incDFM performance in a conventional offline inter-dataset setting: training on one ID dataset (one task) and evaluating once on another OOD dataset. We use CIFAR-10 and CIFAR-100 as ID datasets and SVHN as OOD. We implemented each baseline (DFM, ODIN, Softmax and Mahalanobis) using the same architecture as described in Section 4.1 - a frozen Resnet50 backbone followed by trainable MLP. The latter in the case of methods that perform classification (i.e. ODIN and Softmax). We show that incDFM overperforms the compared baselines as measured by AUROC scores. AUROC stands for area under the receiver operating characteristic curve, which plots the true positive rate (TPR) of in-distribution data against the false positive rate (FPR) of OOD data by varying a threshold. It can be regarded as an averaged score.

5.2 Continual Novelty detection

Intra-dataset OOD, Class incremental novelty detection: Figure 4(a) displays the performance per task of incDFM when evaluated on intra-dataset class incremental novelty detection, shown for CIFAR-10 and CIFAR-100. Additionally, figure 4(b) shows the average performance across tasks for all datasets. We evaluate performance using AUROC and AUPR scores. the latter refers to the area under precision recall curve with respect to the novelty class. Overall,



(a)

	CIFAR-10		iNaturalist		CIFAR-100		EMNIST	
	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR
incDFM	**98.2**	**98.2**	**91.5**	**90.4**	**95.4**	**95.5**	**98.7**	**98.8**
DFM	74.0	72.1	61.3	60.5	63.8	62.2	75.4	71.1
Mahal	75.2	72.2	60.9	61.6	58.7	57.7	67.7	65.0
Softmax	66.6	63.5	70.3	66.8	56.4	52.7	61.1	57.6
ODIN	81.6	79.7	75.3	71.6	53.7	53.2	63.3	60.6

(b)

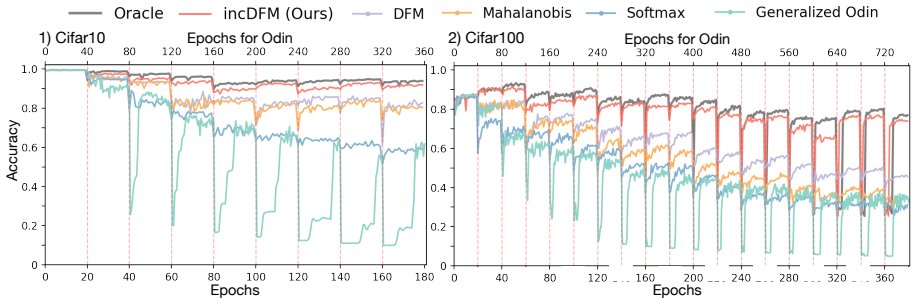
Fig. 4: Intra-dataset Novelty Detection: (a) AUROC scores per task using detected samples for model update. (b) Average AUROC and AUPR scores after all tasks.

our approach over-performs the competing methods. incDFM shows consistent performance over tasks, with minimal to no degradation. We can directly observe the advantage of incDFM's incremental recruitment algorithm by comparing it to DFM (our ablated baseline) which employs a single threshold for OOD selection instead. Additionally, we can observe that the performance gap between incDFM and compared methods is much larger in this class incremental setting then those shown in Table 1 for the offline setting. When ID_t and OOD_t sets are drawn from the same dataset, as is the case in our class-incremental setting, OOD detectors cannot explore low-level statistics to arrive at a prediction. Instead, the distinction must come from more conceptual class-defining properties, arguably harder. Moreover, in this continual setting, other factors such as forgetting and error propagation pose a further challenge.

Inter-dataset OOD, dataset incremental novelty detection: Table 2 shows results for incDFM in a different continual learning setting, where each task now corresponds to a fully novel dataset (experiment described in section 4). In general, all OOD detectors, including incDFM, show higher performance in this experiment than in the previous intra-dataset experiments (refer to Fig 4). This again reaffirms the notion that inter-dataset ID/OOD splits are easier to disentangle than splits within the same dataset. Additionally, we show that baselines Softmax and ODIN really suffer in performance when they don't have access to ground-truth task labels (second row). This finding is in line with other works

Task-Oracle	incDFM		DFM		Mahal		Softmax		ODIN	
	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR
Yes	-	-	-	-	-	-	99.9	99.9	99.8	99.6
No	99.9	99.9	95.0	94.5	94.7	94.3	69.4	70.1	64.2	64.0

Table 2: Inter-dataset continual learning (8-dataset) with and without Task Oracle.



(a)

	Cifar10	iNaturalist	Cifar100	EMNIST
MT	93.1	91.1	75.6	93.1
Oracle	94.0	85.3	77.2	92.8
incDFM	**92.0**	**74.6**	**74.7**	**90.3**
DFM	75.2	60.8	46.2	71.3
Mahal	63.6	55.8	35.6	48.5
Softmax	59.9	62.5	36.2	41.7
Odin	62.5	64.3	36.8	46.1

(b)

Fig. 5: Unsupervised incremental classification pipeline - (a) Average incremental classification accuracy over tasks. (b) Final classification accuracy after all tasks.

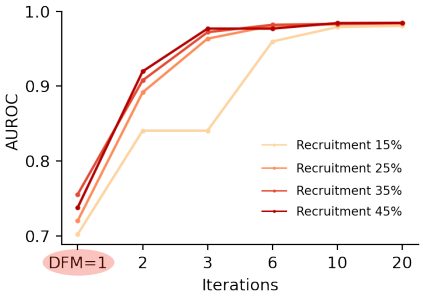
that have explored task oracle substitutions in CL [39]. In fact, having access to task labels for unlabeled ID samples is unrealistic in novelty detection since if a task-label is known, it obviates the need for novelty detection in the first place.

5.3 Full Pipeline Results

Figure 5 shows results for our end-to-end pipeline for unsupervised incremental class learning. In incDFM, the experience replay coreset stores $\widehat{OOD}_t$ samples and their assigned pseudolabels, see section 3.2. Thus, we propose an upper-bound baseline, *Oracle*, which employs the same classifier and experience replay strategy but uses real ground truth novelties(OOD_t) for training and for populating the coreset. This is equivalent to stopping error propagation. We also compare to the multi-task (MT) upper-bound which trains all classes for the dataset jointly, without continual learning. Firstly, figure 5(b) shows that our experience replay baseline using ground-true labels (*Oracle* - dark gray) is reliably close to

AUROC	CIFAR-10		CIFAR-100	
Estimated OOD	Yes	No	Yes	No
incDFM	**98.2**	**98.2**	**95.9**	**96.1**
DFM	74.0	83.0	63.8	65.2
Mahal	75.2	75.8	58.6	58.8
Softmax	66.6	73.8	45.7	67.2
ODIN	81.6	87.3	56.0	69.5

(a)



(b)

Fig. 6: (a) Error Propagation from using $\widehat{OOD}_t$/(yes) vs. ground-truth OOD_t/(no). (b) incDFM iterations and recruitment % (Cifar10 averaged across tasks).

upper bound MT for all datasets, suggesting a consistent mitigation of forgetting through time by using coreset-based replay only. Yet, most importantly, we see that incDFM (red) is very close to the upper-bound *Oracle* for all tasks and datasets, despite using only pseudolabels. In contrast, all other baselines incur a significant drop in classification performance through time. The reason is likely due to the compounded effect of error propagation since they provide a very suboptimal novelty detection performance across tasks (refer back to Figure 4). Poorer $\widehat{OOD}_t$ estimates per task will propagate wrong pseudolabels for training and for coreset storage, adding detrimental noise to the overall training and increasingly hurting performance through time.

5.4 Ablation and Hyper-parameter sensitivity study in incDFM

Error Propagation in continual OOD detection: We analyze the effect of using estimated novel samples, $\widehat{OOD}_t$, versus ground truth novel samples, OOD_t, for knowledge consolidation (Figure 6(a)). Note that estimated $\widehat{OOD}_t$ will contain a degree of error, i.e., ID samples that are erroneously pseudolabeled as novel. Or, instead, too many OOD samples labeled as old. When this error percentage grows too large (as is often the case for hard OOD/ID splits), it begins to detrimentally and progressively affect the ability to perform OOD detection at subsequent tasks. We call this continual compounded effect "error propagation". In incDFM error propagation is largely minimized due to incremental recruitment, which maintains prediction errors low throughout tasks. In contrast, we can see that classification based OOD detectors, e.g. ODIN and Softmax, are particularly vulnerable to error propagation.

Incremental Recruitment sensitivity in incDFM: The number of maximum iterations within a task and the percentage of recruitment of estimated novel remaining samples at each iteration are both hyperparameters in incDFM. We analyze the sensitivity to each in figure 6(b). Note that when maximum iterations is equal to 1 in the x axis, we fall back to single thresholding per

New:Old	incDFM	DFM	Mahal	Softmax	ODIN
1:1	**98.2**	72.1	72.2	63.5	79.7
1:2	**97.0**	56.6	59.5	46.2	62.0
1:3	**95.9**	49.3	52.4	39.3	51.7
1:4	**95.0**	44.2	48.1	34.6	45.1

Table 3: AUPR Scores with task data imbalanced towards more old samples (Cifar10).

task, same as in our ablated baseline DFM. Iterative recruitment seems to peak in performance roughly at about 5 iterations for CIFAR-10 and we observed a similar trend across all datasets. Moreover, performing two iterations is already a 22% improvement when compared to single thresholding as in simple DFM. Alternatively, incDFM is less sensitive to the recruitment percentage and follows an intuitive trend where, for very low recruitment percentages, it takes more iterations to converge (yellow line - 15% recruitment rate). Overall, incDFM with 10 maximum iterations achieves up to a 39.4% improvement over simple DFM.
Mixing Ratio of New/Old in each task: In previous experiments we kept each task with a balanced number of old and new data samples. However, increasing the ratio of old to new data can have a detrimental effect in precision and recall performance. Old classes can be interpreted as distractors and more distractors can make novelty detection harder. We show the effect of data imbalance on performance in Table 3. Overall incDFM is much more robust to imbalances than other baseline methods. From a 1:1 to a 1:4 new to old ratio in the unlabeled pool, incDFM decreases only 3.3% in performance (AUPR scores) whereas baselines have a decrease between 33% (ODIN) to 41.4% (Softmax).

6 Conclusion

This paper presented a novel, self-supervised continual novelty detector. In contrast to the prevailing novelty detection approaches that operate in a static setting, we designed a method capable of handling realistic, non-stationary conditions with recurrent exposure to new classes of data. Using cumulative consolidated knowledge of what is in-distribution up until the new task, our method incrementally estimates a statistical novelty detection model associated to the new task by iteratively recruiting the most certain novel samples and updating itself to progressively enable better estimates. Extensive experimentation in the challenging task-incremental continual learning setting shows state of the art performance in continual novelty detection, minimizing catastrophic forgetting and error propagation at each task through time.

Acknowledgements: This work was supported by the Intel corporation, C-BRIC (part of JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA), DARPA (HR00112190134) and the Army Research Office (W911NF2020053).

References

1. Ahuja, N.A., Ndiour, I.J., Kalyanpur, T., Tickoo, O.: Probabilistic modeling of deep features for out-of-distribution and adversarial detection. In: Bayesian Deep Learning workshop, NeurIPS (2019)
2. 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)
3. Aljundi, R., Reino, D.O., Chumerin, N., Turner, R.E.: Continual novelty detection. arXiv preprint arXiv:2106.12964 (2021)
4. 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)
5. Cheung, B., Terekhov, A., Chen, Y., Agrawal, P., Olshausen, B.: Superposition of many models into one. Advances in neural information processing systems **32** (2019)
6. Cohen, G., Afshar, S., Tapson, J., Van Schaik, A.: Emnist: Extending mnist to handwritten letters. In: 2017 international joint conference on neural networks (IJCNN). pp. 2921–2926. IEEE (2017)
7. De Campos, T.E., Babu, B.R., Varma, M., et al.: Character recognition in natural images. VISAPP (2) **7** (2009)
8. Dhillon, G.S., Chaudhari, P., Ravichandran, A., Soatto, S.: A baseline for few-shot image classification. arXiv preprint arXiv:1909.02729 (2019)
9. Du, X., Charan, G., Liu, F., Cao, Y.: Single-net continual learning with progressive segmented training. In: 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA). pp. 1629–1636. IEEE (2019)
10. 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)
11. French, R.M.: Catastrophic forgetting in connectionist networks. Trends in cognitive sciences **3**(4), 128–135 (1999)
12. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: international conference on machine learning. pp. 1050–1059 (2016)
13. Gallardo, J., Hayes, T.L., Kanan, C.: Self-supervised training enhances online continual learning. arXiv preprint arXiv:2103.14010 (2021)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
15. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks (2017)
16. Hendrycks, D., Mazeika, M., Dietterich, T.: Deep anomaly detection with outlier exposure. In: International Conference on Learning Representations (2019)
17. Hsu, Y.C., Shen, Y., Jin, H., Kira, Z.: Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10951–10960 (2020)
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

19. 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)
20. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
21. Kwon, G., Prabhushankar, M., Temel, D., AlRegib, G.: Novelty detection through model-based characterization of neural networks. In: IEEE International Conference on Image Processing. pp. 3179–3183 (2020)
22. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: Advances in neural information processing systems. pp. 6402–6413 (2017)
23. Lee, K., Lee, K., Lee, H., Shin, J.: A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In: Advances in Neural Information Processing Systems. pp. 7167–7177 (2018)
24. Liang, S., Li, Y., Srikant, R.: Enhancing the reliability of out-of-distribution image detection in neural networks (2018)
25. Lopez-Paz, D., Ranzato, M.: Gradient episodic memory for continual learning. Advances in neural information processing systems **30** (2017)
26. Maji, S., Rahtu, E., Kannala, J., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. arXiv preprint arXiv:1306.5151 (2013)
27. Mallya, A., Lazebnik, S.: Packnet: Adding multiple tasks to a single network by iterative pruning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 7765–7773 (2018)
28. Mundt, M., Hong, Y.W., Pliushch, I., Ramesh, V.: A wholistic view of continual learning with deep neural networks: Forgotten lessons and the bridge to active and open world learning. arXiv preprint arXiv:2009.01797 (2020)
29. Ndiour, I., Ahuja, N.A., Tickoo, O.: Out-of-distribution detection with subspace techniques and probabilistic modeling of features. arXiv preprint arXiv:2012.04250 (2020)
30. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning (2011)
31. 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)
32. Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al.: Conditional image generation with pixelcnn decoders. In: Advances in neural information processing systems. pp. 4790–4798 (2016)
33. Parisi, G., Kemker, R., Part, J., Kanan, C., Wermter, S.: Continual lifelong learning with neural networks: A review. Neural Networks (2019)
34. Petrov, A.A., Dosher, B.A., Lu, Z.L.: The dynamics of perceptual learning: an incremental reweighting model. Psychological review **112**(4), 715 (2005)
35. Quattoni, A., Torralba, A.: Recognizing indoor scenes. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 413–420. IEEE (2009)
36. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 2001–2010 (2017)
37. Ren, J., Liu, P.J., Fertig, E., Snoek, J., Poplin, R., Depristo, M., Dillon, J., Lakshminarayanan, B.: Likelihood ratios for out-of-distribution detection. In: Advances in Neural Information Processing Systems. pp. 14707–14718 (2019)
38. Rios, A., Itti, L.: Closed-loop memory gan for continual learning. arXiv preprint arXiv:1811.01146 (2018)

39. 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)
40. Rolnick, D., Ahuja, A., Schwarz, J., Lillicrap, T., Wayne, G.: Experience replay for continual learning. Advances in Neural Information Processing Systems **32** (2019)
41. Shin, H., Lee, J.K., Kim, J., Kim, J.: Continual learning with deep generative replay. Advances in neural information processing systems **30** (2017)
42. Sun, J., Yang, L., Zhang, J., Liu, F., Halappanavar, M., Fan, D., Cao, Y.: Gradient-based novelty detection boosted by self-supervised binary classification. arXiv preprint arXiv:2112.09815 (2021)
43. Van Horn, G., Cole, E., Beery, S., Wilber, K., Belongie, S., Mac Aodha, O.: Benchmarking representation learning for natural world image collections. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12884–12893 (2021)
44. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011)
45. 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)
46. Wu, C., Herranz, L., Liu, X., van de Weijer, J., Raducanu, B., et al.: Memory replay gans: Learning to generate new categories without forgetting. Advances in Neural Information Processing Systems **31** (2018)
47. Yoon, J., Yang, E., Lee, J., Hwang, S.J.: Lifelong learning with dynamically expandable networks (2018)
48. Zeng, G., Chen, Y., Cui, B., Yu, S.: Continual learning of context-dependent processing in neural networks. Nature Machine Intelligence **1**(8), 364–372 (2019)