HAL
open science

Novelty detection for unsupervised continual learning in image sequences

Ruiqi Dai, Mathieu Lefort, Frédéric Armetta, Mathieu Guillermin, Stefan Duffner

To cite this version:

Ruiqi Dai, Mathieu Lefort, Frédéric Armetta, Mathieu Guillermin, Stefan Duffner. Novelty detection for unsupervised continual learning in image sequences. IEEE International Conference on Tools with Artificial Intelligence (ICTAI), Nov 2021, Washington DC (virtual), United States. 10.1109/ICTAI52525.2021.00080 . hal-03465146

HAL Id: hal-03465146

https://hal.science/hal-03465146v1

Submitted on 3 Dec 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Novelty detection for unsupervised continual learning in image sequences

Ruiqi Dai*§, Mathieu Lefort†§, Frédéric Armetta†§, Mathieu Guillermin‡ and Stefan Duffner*§
*Université de Lyon, INSA Lyon
†Université de Lyon, Université Claude Bernard Lyon 1
‡Université Catholique de Lyon, France
§LIRIS, UMR 5205 CNRS, France

***Abstract*—Recent works in the domain of deep learning for object recognition on common image classification benchmarks often address the representation learning problem under the assumption of i.i.d. input data. Although achieving satisfying results, this assumption seems not realistic when agents have to learn autonomously. An autonomous agent receives a continual visual flow of objects which is far from an i.i.d. distribution of objects. Moreover, agents have to construct their representations of the world and adapt to unknown environments, without relying on external sources of information such as labels that would be provided post-classification and are unavoidable when an over-segmentation is done. Then, in order to exploit the learned representation effectively for object recognition, a clear and meaningful relationship w.r.t. real object categories is required, which has been largely neglected in existing unsupervised algorithms.**

In this paper, we propose a novelty detection method for continual and unsupervised object recognition, as an extension for the recent CURL model, which allows to moderate over-segmentation while preserving accuracy, in order to meet the requirements for autonomy. We experimentally validated our approach on two standard image classification benchmarks, MNIST and Fashion-MNIST, in this unsupervised and continual learning setting and improve the state of the art in terms of cluster purity, which is crucial for subsequent object recognition, since it facilitates clustering when information on ground truth labels is not available for free.

***Index Terms*—Continual learning, class-incremental learning, novelty Detection, object recognition, unsupervised learning.**

I. Introduction

Let's consider an agent interacting with objects in an unknown environment, continuously perceiving the objects through sensors. Being able to adapt to changes in the environment as well as continuously building a (visual) representation of objects of new classes while exploiting acquired knowledge is a crucial property for such a dynamic machine learning system. Classical deep learning models have shown excellent performance on image classification in an "off-line" setting, making the learning scenario comparatively simple in terms of representation learning since an iterative stochastic optimization of the loss function on i.i.d. data can be applied efficiently. However, when training data is not available all at once but sequentially, these models face some severe limitations. The literature on continual learning with neural networks [23], [24] partially responds to this issue, yet many of them are supervised (i. e. supervised learning) and in order to effectively classify new observations of learned objects, extensive class labels are needed either at training time or after training to correctly attribute the numerous learned clusters to meaningful object categories. Even when addressed specifically, the lack of control over the way objects are introduced to the system lead to catastrophic forgetting phenomena for objects not seen for a long time, which is still a limitation to be overcome and considerably decreases the clustering performance. In fact, being able to learn new knowledge is an advantage coming from the plasticity of the network, but at the same time, the network should be stable enough to maintain the acquired knowledge, according to the stability-plasticity dilemma. A further limitation that we could identify, in the case of online and unsupervised learning, is the tendency to oversegment categories into many additional clusters [4], which makes the grouping of clusters inefficient during evaluation. Indeed, even if the clustering is done following an unsupervised approach, its evaluation is generally done thanks to ground truth labels assigned to the generated cluster, but hardly available online, which makes the methods ineffective when seeking autonomy. Moreover, we underline that over-segmentation facilitates the achievement of good accuracy while paradoxically reducing the autonomy of the system.

In this paper, we address the problem of unsupervised class-incremental representation learning for object recognition, in which objects are observed one after the other for a single period of time without storing any images in the long term. We will name it as *class-incremental* learning in the following sections. We propose a deep neural network model performing unsupervised class-incremental learning for visual object recognition, which is an extension of CURL [23], an approach dedicated to continual learning. Our main contribution lies in the integration of a new-class estimator based on statistics of the dynamics of the input sequence leveraging the temporal continuity of objects introduced and allowing to improve the detection of new objects. Furthermore, this guides the training with self-supervision by optimizing an adapted loss function.

The re-identification of learned objects that re-appear at different times during training could be addressed by using a classifier for category prediction which may require dedicated mechanisms. We choose to set aside this problem that can be addressed on its own as a second step. In this paper, we focus on the problem of performing an accurate automatic and

unsupervised novelty detection, in order to maintain clustering as close as possible to the original class labels provided by the dataset while keeping high accuracy.

II. Related work

A. Novelty Detection

In the literature, there are two types of tasks considering novelty detection: one-class classification approaches [19] that consider novelty detection as a binary classification problem of known/unknown, which are limited in scalability when there are numerous categories in the dataset; or multi-class approaches, also called open-set classification in the literature [1], [6]. For multi-class novelty detection, the estimation of the probability for unknown objects is a major challenge because existing classification approaches are usually based on a closed-world assumption [1], which estimates the probability distribution only over known categories, thus does not provide an appropriate estimation of the uncertainty when it comes to an unknown object. As a result, the model may wrongly "activate" an existing category with high confidence [16], this creates calibration problems in commonly used classification approaches using the softmax function. Some proposed approaches re-calibrate softmax, for example, ODIN [15] or G-OpenMax [5], [6]. Others use ensembles of deep learning models to predict uncertainty [12]; or treat this issue with a probabilistic approach based on the likelihood ratio between the inlier distribution and background knowledge [25], [28]. However, in continual learning, it is much more challenging to have a precise estimate on the background statistics for this sort of calibration.

B. Unsupervised object recognition

Different approaches for unsupervised image classification have been proposed in the literature, contrary to continual learning, common "off-line" approaches assume that training data are i.i.d [8], [30]. It is usually necessary to present the entire dataset several times in random order during training to ensure convergence and optimal performance. Recent advances in this domain make use of deep neural networks, in particular generative models [2], [7], [10] like Variational Auto-Encoders (VAE) [10], [31] and Generative Adversarial Networks (GAN) [2], [7]. These models learn to generate new data with the same statistics as a training set. Another family of unsupervised object recognition concerns clustering approaches like k-means [27] or DBSCAN [33]. These approaches work on the raw data without learning high-level features as deep neural networks, so applying them to images requires to use "hand-crafted" local feature extractor. Others include incremental clustering, for example SOINN [4] that learns the topology of dataset distribution, which will be introduced more in detail in section II-C. Common unsupervised object recognition algorithms have difficulties in determining the number of categories, as a result, they tend to mix similar categories, or reversely divide a category into several subcategories, requiring an extra effort of regrouping clusters during evaluation.

C. Continual learning

The literature in continual learning concerns two different scenarios: either solving a sequence of tasks/learning different datasets in the multi-task scenario, or learning new classes [18] incrementally in the single-task scenario. The state-of-the-art methods for continual learning with neural networks [14], [22] mainly focused on 3 categories of approaches:

- ***Structural approaches*** propose approaches that is network structure-related, for example, [26] proposes to dynamically add new nodes during training that modify the network structure with respect to the arrival of new tasks. Other algorithms [17] selectively activate parts of the network.
- ***Regularization approaches*** add a task-related regularization term to the cost function [34] to moderate changes in neurons involved in previous tasks while still allowing the network to learns new tasks. For example, in [11] the effect of catastrophic forgetting is contained by constraining the update of weights via a regularization term based on the Fisher information matrix extracted from previous tasks.
- ***Experience replay*** approaches try to alleviate catastrophic forgetting by regularly "replaying" past training examples [24], i. e. to train with both images from the current task/class and stored or generated samples [23]. The strategy of replay or the choice of examples to be stored is crucial to the model in terms of memory efficiency.

Most of these approaches are designed for *supervised* continual learning, showing strong dependence on accurate task identification and instance ground truth labels. Concerning unsupervised continual learning, the Self-Taught Associative Memory (STAM) [29] is an approach based on hierarchies of clustered image features that are continually learned by selecting centroids based on distance metrics. However, as opposed to neural network-based models, it is not clear to what extent the learned representation (i. e. hierarchical sets of image patches) can generalise to unseen object appearances and can be "re-used" for new object categories. Continual Unsupervised Representation Learning (CURL) [23] proposed a model based on VAE learning a Gaussian Mixture for different categories and alleviates catastrophic forgetting with generative replay, but it fails to automatically detect the number of clusters, thus requires to group clusters during evaluation. Therefore, *unsupervised* continual learning remains a challenging open research problem.

Common incremental clustering methods [3], [9] (such as BIRCH [35], incremental k-means [3]) are potential approaches to address incremental learning. Other approaches make use of topology learning [4]. Furao et al. [4], for example, proposed a model called SOINN, for unsupervised and online topology learning for non-stationary data, with less memory consumption and allows for learning without knowing *a priori* the number of classes and the distribution of data. Yet in the domain of image sequences, to work effectively with more complex visual data streams, these approaches often

require either hand-crafted features or a pre-trained feature extraction model. Comparatively, approaches based on deep learning are more suitable due to the powerful representation capacity for visual data and images. Another limitation of these approaches is that they tend to create a large number of clusters and thus "over-segmenting" the original object classes [4]. This is also the case for some of the unsupervised continual learning approaches mentioned previously, cf. CURL [23]. This makes subsequent classification more complex as supervision is required afterwards to assign each cluster to the corresponding object class.

We propose a model that improves the clustering effectiveness by exploiting the constraints of the addressed scenario where objects are presented one after the other in the data stream. Our model is based on a previously proposed generative deep neural network [23] that we extended by modifying and improving the loss function and the new object class detection process.

III. Proposed Approach

Regarding the context and the class-incremental setting (cf. section I) of unsupervised and continual learning of object representation, we propose a generative neural network model extending CURL [23] that has been originally designed for the single-task sequential learning. We will first briefly outline this base model in section III-A, and then present our contributions (sections III-B and III-C).

A. Model and learning algorithm

CURL is a model that learns robust representations for different classes in a continuous manner based on a derivative of Variational Auto-Encoder (VAE), as shown in Fig. 1. Concretely, the core of the model is a Variational Auto-Encoder (VAE) which allows to approximate the distribution of the latent variable with a Gaussian or component. CURL extends the VAE by dynamically introducing a new dedicated component, for each new outlier image. It alleviates the effect of catastrophic forgetting by continuously generating synthetic training examples of previously learnt classes.

The model optimizes a modified ELBO (Evidence Lower Bound) objective (maximizing the likelihood of the data), with input images x, categorical variable y (the index of the Gaussian component), latent variable z corresponding to the internal representation (formed by the GMM):

$$E(x) = \sum_{n=1}^{K} q(y=k|x)\Big[\log p(x|\widetilde{z}^{(k)}) - KL((q(z|x,y=k)||p(z|y=k))\Big] - KL(q(y|x)||p(y)) \ , \tag{1}$$

where $q(y=k|x)$ represents the component posterior, computed by a dense layer with softmax, marked as yellow nodes in Fig. 1, $\widetilde{z}^{(k)} \sim q(z|x,y=k)$ is the latent code sampled from the kth Gaussian component each modelled by a dense layer of latent encoder head, $log\ p(x|\widetilde{z}^{(k)})$ corresponds to the component-wise reconstruction loss of input images,

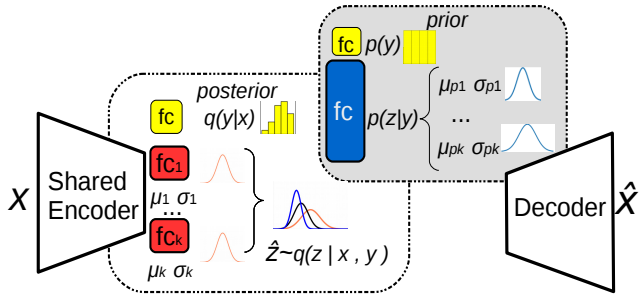

Fig. 1. The neural architecture of CURL: a Variational Auto-Encoder, X representing the input images, y the category variable. The encoder maps the input images to a shared representation for all the categories. Its output is used as the input of a fully-connected (fc, in yellow) softmax layer to estimate the object category $q(y|x)$; and updates the parameters μ_k, σ_k of the corresponding component(s) k. Posterior latent variable distribution $q(z|x,y)$ is approximated with component-specific latent encoders (a GMM). Also, the prior $p(z|y)$ of latent variable z follows a Gaussian distribution. Then, the image $\hat{X}$ is reconstructed from the resampled $\hat{Z}$ using the decoder. For more détails, refer to section III.

with reconstructed image $\hat{X}$ at the output, $KL((q(z|x,y=k)||p(z|y=k))$ is a Kullback-Leibler divergence acting as the component-wise regularizer and enforcing a (Gaussian) embedding and $KL(q(y|x)||p(y))$ is the categorical regularizer that ensures that classes are well balanced, approaching the assumed uniform prior distribution $p(z|y)$ of each category. By maximizing Eq. 1, the model learns to reconstruct the input images and at the same time, due to the two regularization terms, to cluster objects into different classes in the latent space z by dynamically assigning them to different components. Poorly modelled instances whose ELBO is inferior to a threshold are considered as possible new category candidates and are thus stored in a temporary buffer which, once filled, is used to create and initialize a new component in the model. For more details, refer to [23].

This base model gives promising clustering performance. However, a major limitation is that the number of clusters resulting from the model does not allow to stay close to the original number of classes. In fact, it tends to create excessive clusters that therefore do not reflect the actual distribution of object categories. The number of introduced clusters is expected to stay close to the ground truth distribution, to facilitate eventually the categorization during evaluation.

Due to the fact that objects are presented sequentially for a certain amount of time, we consider that continuity is present in the perception of an autonomous agent evolving in a continual environment. Thus, the purpose of this study relies on measuring the additional value allowed by considering such a hypothesis on the accuracy of the system, training time and the need to keep under control the number of introduced components.

We propose two modifications of the original CURL model: a new category detection process (section III-B) that guides the learning with self-supervision optimizing a modified loss function (section III-C).

B. Detection of new classes

We hereby consider the case where the agent perceives objects class by class in the environment, not in a completely random way but in a class incremental way as it is mentioned in section I. In this paper, we choose to focus on novelty detection and improvements that can be achieved through the use of the continuity of perceived objects hypothesis, illustrated by the continuity in classes presented to the system. This shows the potential for such an approach, which is generally not exploited in machine learning, to study later the resilience of the process in a more noisy environment.

In this context, our contribution consists in the automatic detection of new classes by integrating an adaptive change detection algorithm, the Page-Hinckley test [21] applied to ELBO likelihood, a common approach applied in the domain of concept drift detection to detect abrupt changes in sequential input data. Formally, let $x_t \in \mathcal{X} = \{x_0, \ldots, x_T\}$ be the examples presented in sequence of input training examples. In accordance with CURL, in our model, poorly modelled examples, are considered as new category candidates, i. e. for which the *unsupervised* ELBO objective $E(x)$ (Eq. 1) is below a threshold θ, since the *unsupervised* ELBO objective $E(x)$ marginalizes over all the existing categories which might reduce false-positive new category detection that corresponds to a category learned in the past instead of a new one. In our model, we apply the Page-Hinckley test that computes the decision function $g(t)$ for each new arriving example. We compare ELBO objective $E(x)$ with a threshold θ, noted by H the Heaviside step function that will equal to 1 if $E(x)$ is smaller than a threshold θ (implying an outlier). It is smoothed by a running average noted by $p_n(t)$, counting the average times that the outliers occur. We adopted a variant of the Page-Hinckley test as defined in [20], with N being the number of samples the agent has seen since the previous category change, and υ being the tolerated change for each step:

$$g(t) = \max(0, g(t-1) + p_n(t) - \mu_{p_n}(t) - \upsilon) \tag{2}$$

$$\mu_{p_n}(t) = \frac{(N-1)}{N}\mu_{p_n}(t-1) + \frac{1}{N}p_n(t) \tag{3}$$

$$p_n(t) = \alpha * p_n(t-1) + (1-\alpha) * H(\theta - E(x_t)) \ . \tag{4}$$

If $g(t)$ is greater than a threshold θ_n, then a new category is detected, i. e. a Gaussian is added to the GMM in the VAE and we reinitialize $g(t)$ to 0. Contrary to CURL that might be affected by noise in the ELBO loss, under the hypothesis of temporal continuity, our proposal of detecting new categories by Eq. 2-Eq. 4. helps to smooth these fluctuations and to obtain a cleaner supervision signal in the presence of outliers and alleviate category "over-segmentation".

Another modification of CURL in our model is that we propose for the original CURL model concerns the usage of the buffer storing recent examples in the incoming data stream. In our model since the proposed Page-Hinckley test detects abrupt changes, once a category change is captured, the buffer is filled with all the following instances in the sequence until reaching its maximum size n. However, the examples in the (unfilled) buffer are not used for training immediately to prevent over-fitting resulting from too few training instances and to ensure having enough observations for each object class. Once the buffer is full, the training of the new class is initiated and the buffer is released.

C. Loss function

We use self-supervision deduced from our new-category detection algorithm to adapt the loss function that is used for training the model. We propose to optimize a supervised version of the ELBO objective function $E_{sup}(x)$ that CURL [23] originally used for a supervised baseline comparison of their algorithm. However, we integrate it differently in our approach. That is, we create an internal supervision signal $y_m \in \mathbb{N}$ based on the detection of new classes for training. $y_m \in \mathbb{N}$ that corresponds to the class of the instance determined by our model. Note that our proposed approach is still completely unsupervised as no ground truth labels are used. More specifically, y_m is incremented if the presence of the new, unseen object class is detected and maintained constant otherwise.

$$y_m = \begin{cases} y_m + 1, & \text{if } g_t \geq \theta_n \\ y_m, & \text{otherwise.} \end{cases}$$

The objective is defined as:

$$\begin{aligned} E_{sup}(x) = \log q(y = y_m|x) + \log p(x|\widetilde{z}^{y_m}, y = y_m) \\ - KL(q(z|x, y = y_m)||p(z|y = y_m)) \ , \end{aligned} \tag{5}$$

and we continue to use the same variable definition as in Eq. 1. where the first term trains a fully connected layer with softmax to predict the label, the second term minimises the auto-encoded reconstruction error and the last term again represents the Kullback-Leibler divergence between the variational posterior of z and its corresponding Gaussian prior distribution.

IV. Experiments

A. Dataset

To compare our approach to the state of the art, we evaluated our model on two standard datasets: MNIST (images of handwritten digits from 0 to 9) [13] and Fashion-MNIST [32] (Zalando's images with the classes {T-shirt, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, ankle boot}). Both datasets contain objects from 10 classes, with 60000 images for training and 10000 images for testing. The size of images in both datasets is 28×28. For each class, there are around 6000 images for training and around 1000 for the test set. During training we present images in a class by class order, from 0 to 9. It can be noticed that compared to the MNIST dataset, the Fashion-MNIST dataset is more complex. In the Fashion-MNIST dataset, images of different classes can be more similar (than in MNIST), for example, dresses resemble coats. The MNIST dataset, however, is a comparatively simple task for the reason of the well-alignment of digits in each category. And comparatively, objects in the Fashion-MNIST dataset are more diverse and more complex.

B. Experimental protocol

Our model is trained in a continuous way, as stated in section I, i. e. the data are presented to the model sequentially and class-by-class. Thus, each class is seen only once, but during training, each image of the current class can be presented several times until a new class is detected. For a fair comparison with CURL, as detailed in section IV-C, we preserved the model architecture and setting for generative replay from [23], meanwhile tuning the other parameters (the threshold for outliers) with respect to clustering performances, and most importantly, with regard to the optimum number of categories detected that approaches the true distribution of categories and results in the best clustering score of AMI/ARI while preserving the same performance (i. e. accuracy) in clustering. We also compared the clustering performance of our proposal with two incremental clustering algorithms, BIRCH [35] and incremental k-means [27]. The two mentioned incremental algorithms do not provide explicit feature extracting methods, to avoid retraining a neural network, we take the flattened image as input without extracting features. In addition, we compare our model with SOINN [4], which was originally designed for offline clustering, that we adapted in an online setting that presents objects class by class. As an ablation study, we have also tested two variants of our model: Ours w/o P-H test, where we perform a simple running average p_n while comparing ELBO objective with a threshold; Ours w/o p_n, where we apply the Page-Hinckley test directly on ELBO objective without running average p_n smoothing.

C. Hyperparameters

For both datasets, we fixed the neural network architecture and the learning rate to 10^{-3} while using the Adam optimizer. To compare our model with CURL, we use the same neural network structure as in [23]: a 4-layer MLP as encoder {1200, 600, 300, 150}, and a linear layer with 64 dimensions to compute the mean and variance for the 32-dimensional latent variable z. For the decoder a two-layer MLP {500, 500} was used. The total number of iterations is 100000 counting all the categories, for each category 10000 iterations, where at each iteration, the size of batch is 100. We applied the mechanism of generative replay in the same way as CURL, i. e. images of previous classes are generated at fixed intervals (every 10000 steps) and stored into a buffer. For the mixture generative strategy, we continue to use the one of CURL, that is to create a mixture between real images of the current category and generated images of other learned components. To this end, every two steps, a batch of generated images is mixed with the batch of real images for training. We suppose that images of a class are visible for at least 100 steps in both experiments and we use a buffer of size 100 that stores outlier candidates. For the value of θ, the threshold on the ELBO loss, we have chosen $\theta = -150$ for all the experiments on MNIST, resulting in the best accuracy. For Fashion-MNIST, we set $\theta = -300$ for CURL and our model without Page-Hinckley test, and $\theta = -190$ for our model. Concerning the Page-Hinckley test, we set $\alpha = 0.85$ for both datasets, and $\upsilon = 0.3$ and $\theta_n = 1.5$ for parameters in the experiments with Page-Hinckley test applied on running average p_n; $\upsilon = 55.0$ and $\theta_n = 1500.0$ for parameters in the experiments with Page-Hinckley test applied on *negative* unsupervised ELBO loss without p_n smoothing.

D. Evaluation measures

To evaluate the quality of the learned clustering, we used three standard metrics: the clustering accuracy assigning to each component its most represented class, for labelisation of each component on the test set in correspondence to classes and measuring the proportion of correctly classified instances, the Adjusted Mutual Information (AMI) and the Adjusted Random Index (ARI) computed between learned clustering prediction and that of the ground truth. AMI measures the mutual information between two assignments of partitions. ARI measures the similarity between two partitions by counting the difference of assignment of pairs of samples between two partitions. Both metrics are adjusted w. r. t. the chance to remove the bias induced by the inequality in the number of clusters in both partitions. All measures are in $[0, 1]$, where higher values are better.

The clustering accuracy gives a general idea about the classification performance if labels were available. However, it does not completely reflect the quality of the clustering. For example, let's consider the case where the algorithm creates a partition that correctly separates different classes, but creates many excessive clusters from the same class (over-segmentation). We need at least one ground-truth label per cluster to regroup them into correct classes, i. e. requiring supplementary effort on data annotation, which considerably decreases the level of autonomy of the algorithm in an unsupervised continual learning setting.

E. Results

The results on MNIST and Fashion-MNIST are shown in Table I and in Table II respectively. Note that one needs to choose the trade-off between optimizing the number of clusters, reaching better AMI/ARI scores, while detecting all the changes which allows high clustering accuracy. For MNIST, our model achieves a very good trade-off and creates fewer additional components, i. e. the closest to the real number of classes (10), and scores the highest in terms of AMI and ARI compared to CURL and SOINN. For Fashion-MNIST, our model outperforms CURL on the AMI and ARI measures, with a slightly inferior accuracy. But as shown in Table II, CURL creates 120 components exceeding by far the number of real categories in the dataset. This indicates that the clusters created by our model follow the true distribution of different categories and avoid over-segmentation.

In Fig. 2 and Fig. 3, we further show the confusion matrix between ground truth classes and clusters. We can observe that in our model, samples of the same class are represented principally by one cluster. On the contrary, the confusion matrix of CURL shows that CURL tends to separate samples of the same class into different clusters. We equally illustrate the

Model	accuracy	AMI	ARI	nb components
CURL [23]	0.822 ± 0.0102	0.557 ± 0.006	0.28 ± 0.025	93.85 ± 1.884
CURL supervised [23]	0.855 ± 0.006	$\mathbf{0.749 \pm 0.006}$	0.6997 ± 0.012	$\mathbf{10 \pm 0}$
SOINN [4]	$\mathbf{0.925 \pm 0.0011}$	0.39 ± 0.002	0.018 ± 0.0008	1204 ± 39.6
BIRCH [35]	0.3026 ± 0.002	0.184 ± 0.014	0.10 ± 0.0113	$\mathbf{10 \pm 0}$
Increm. k-means [27]	0.338 ± 0.017	0.2545 ± 0.013	0.124 ± 0.013	$\mathbf{10 \pm 0}$
Ours w/o P-H test	0.849 ± 0.008	0.735 ± 0.0102	0.685 ± 0.015	22 ± 1.07
Ours w/o p_n	0.854 ± 0.005	$\mathbf{0.748 \pm 0.00424}$	$\mathbf{0.6996 \pm 0.0085}$	10.67 ± 0.47
Ours	0.8537 ± 0.006	$\mathbf{0.746 \pm 0.0096}$	$\mathbf{0.70 \pm 0.013}$	$\mathbf{10 \pm 0}$

TABLE I
COMPARISON OF OUR METHOD WITH THE STATE OF THE ART ON THE MNIST (AVERAGE OVER 3 RUNS) FOR EACH METRIC MEAN±SD.

Model	accuracy	AMI	ARI	nb components
CURL [23]	0.686 ± 0.013	0.445 ± 0.004	0.137 ± 0.002	120 ± 0.0
CURL supervised [23]	0.654 ± 0.007	$\mathbf{0.57 \pm 0.006}$	0.4336 ± 0.006	$\mathbf{10 \pm 0}$
SOINN [4]	$\mathbf{0.796 \pm 0.003}$	0.365 ± 0.001	0.022 ± 0.008	755 ± 16.54
BIRCH [35]	0.328 ± 0.023	0.286 ± 0.019	0.124 ± 0.0139	$\mathbf{10 \pm 0}$
Increm. k-means [27]	0.404 ± 0.004	0.38 ± 0.0105	0.237 ± 0.013	$\mathbf{10 \pm 0}$
Ours w/o P-H test	0.644 ± 0.009	0.537 ± 0.015	0.415 ± 0.022	64.6 ± 9.5
Ours w/o p_n	0.65 ± 0.0098	0.547 ± 0.007	0.42 ± 0.007	25.67 ± 9.534
Ours	0.6526 ± 0.0056	0.558 ± 0.00856	$\mathbf{0.442 \pm 0.0117}$	13.0 ± 2.19

TABLE II
COMPARISON OF OUR METHOD WITH THE STATE OF THE ART ON FASHION-MNIST (AVERAGE OVER 3 RUNS) FOR EACH METRIC MEAN±SD.

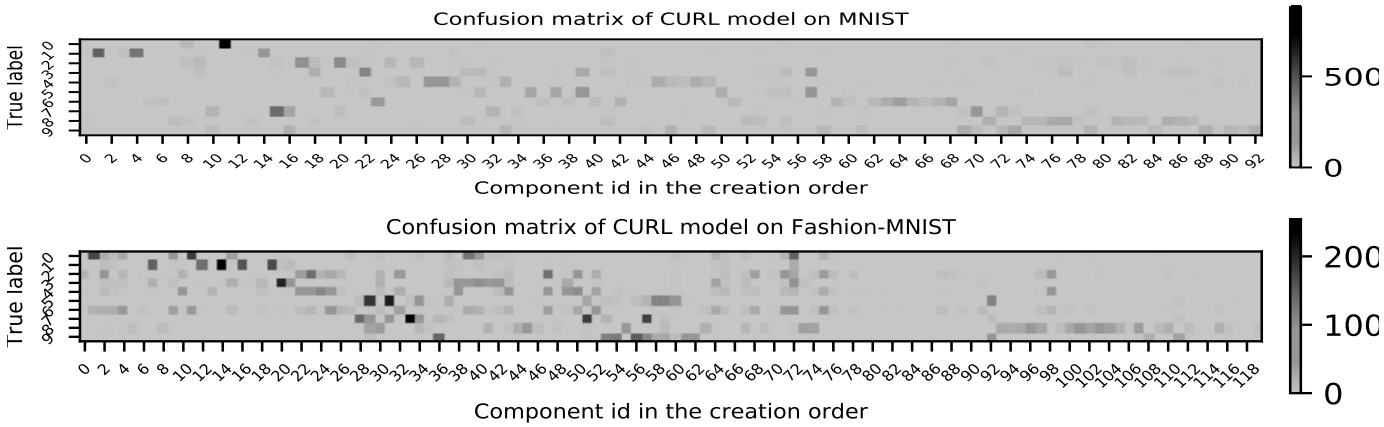

Fig. 2. Confusion matrices between ground truth and predicted cluster components for CURL on the MNIST test set (above) and the Fashion-MNIST test set (below) (the darker a cell the more instances it represents).

2D t-SNE projection of the learned embedding vector of our model and CURL on the MNIST test set in Fig. 4. Different colors represent different categories according to the ground truth label. This not only shows that our approach reduces the phenomenon of over-segmentation in the clustering but also that the different clusters are more consistent with the real object classes. In addition, the clusters are overall more compact and better separated.

Finally, we explicitly studied the relationship between clustering accuracy and the amount of available annotated training data during evaluation, as shown in Fig. 5. We illustrated the variation of clustering accuracy, while using a limited number of examples on the test set to attribute the majority class to each component. Examples used for labeling were chosen at random and with a permutation at each evaluation. Compared to CURL, our model can achieve its maximum accuracy with a very small amount of labelled examples during evaluation, while CURL requires much more examples. The over-segmentation clearly increases the requirement of annotated data during evaluation and may thus limit the classification performance in practical applications.

To validate the individual contributions of our method, we compared it to a variant of CURL using our loss (Eq. 5) supervised by the ground truth and with buffer, called *"CURL supervised"* in Tables I and II. These two experiments demonstrate the effectiveness of our new-category detection algorithm, since our model with Page-Hinckley test applied on p_n is capable of reaching a comparable performance in

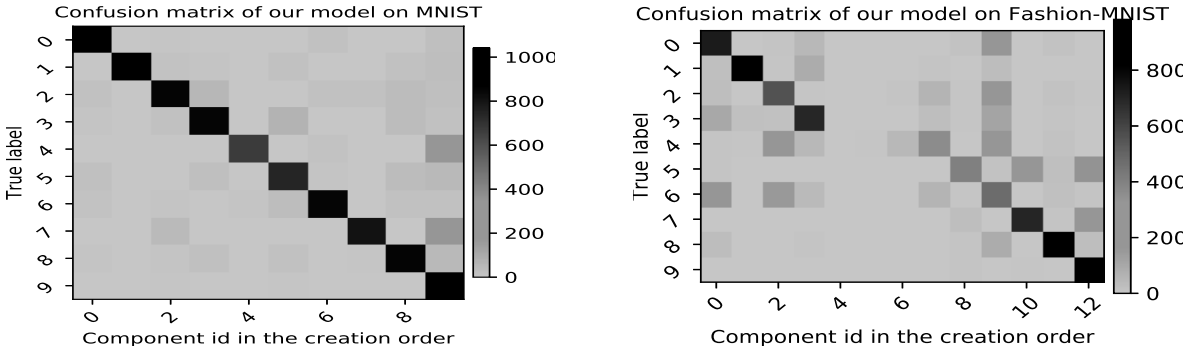

Fig. 3. Confusion matrices between ground truth and predicted cluster components for our complete approach on MNIST (left) and Fashion-MNIST (right) (the darker a cell the more instances it represents).

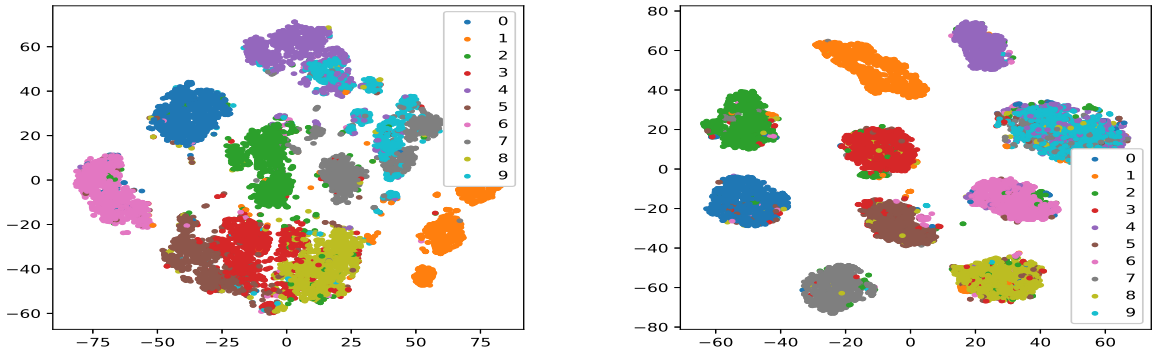

Fig. 4. 2D t-SNE projection of one run of CURL (left) and our model (right) on the MNIST test set: different colors represent different ground truth labels. There are about 93 clusters in CURL representing 10 categories, thus several "sub-clusters" for each category

terms of new category detection on MNIST with respect to supervision using the ground truth.

But both our model and CURL are outperformed by SOINN in terms of clustering accuracy. Only in terms of accuracy, SOINN performs better, which is not surprising given the excessive number of components (about 1200 on MNIST and about 755 nodes on Fashion-MNIST) reducing thus the probability of impure ground-truth clusters but at the same time needing much more additional supervision to label these clusters, as demonstrated in Fig. 5, one could observe that if we only use part of the test set to label components by their majority class, a drop in the clustering performance could be remarked from Fig. 5. The SOINN model converges the slowest compared to CURL and our model.

The results of BIRCH and incremental k-means are much below the performance of the other methods on both datasets showing a clear limitation of such classical incremental clustering algorithms in this context.

V. Conclusion and Discussion

Recent works have focused on creating an efficient neural network model for continual learning, as it is the case for CURL which is unsupervised and provides a generative replay mechanism while making use of a rich multivariate Gaussian Mixture Model. In this paper, we improved the new category detection process by moderating the number of components created for class categorization in order to stay close to the real distribution. We consider that, for an autonomous agent, some continuity is present and images of its environment are not perceived in a totally random order. Thus, we proposed a completely unsupervised approach based on an extension of CURL, a VAE-based model [23], that takes advantage of continuity in the introduced object class and applying a supervised ELBO loss with self-supervision. To this end, we proposed to use the statistical Page-Hinckley test to improve the performance of new-class detection, and p_n a running average for each instance, to smooth fluctuations in the ELBO loss, leading to a robust class change detector. When compared to the baseline, our proposal allows to considerably reduce the introduction of additional clusters while keeping accuracy, which improves autonomy. Indeed, over-segmentation of clusters leads to further supervision for classification which is not always available online, or can only be done in a restrained way. This work appears as a first step and shows how unsupervised learning can take advantage of temporal continuity of objects perceived to better categorize objects online. Further work will study how this proposal behaves under increasing noise in the input sequences.

References

[1] Bendale, A., Boult, T.E.: Towards open set deep networks. In: CVPR. pp. 1563–1572 (2016)

[2] Bojanowski, P., Joulin, A., Lopez-Pas, D., Szlam, A.: Optimizing the latent space of generative networks. In: International Conference on Machine Learning. pp. 600–609 (2018)

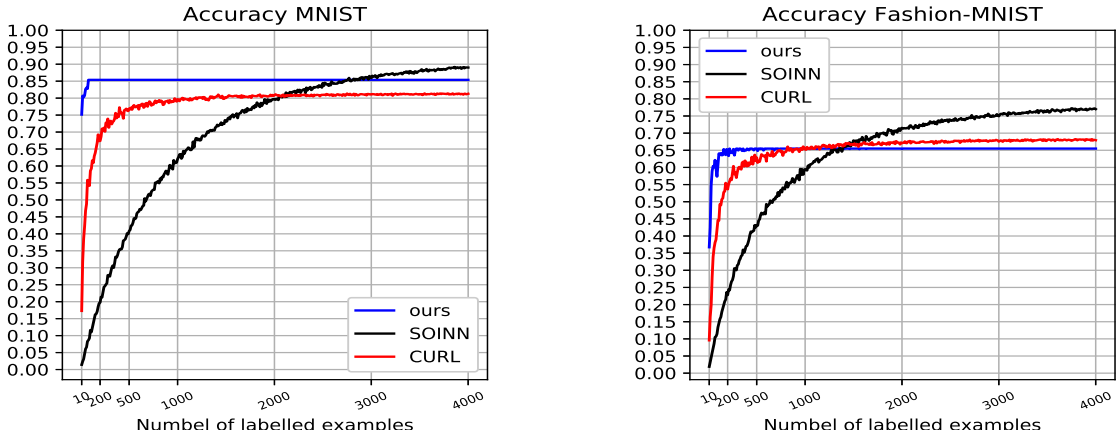

Fig. 5. Influence of of the number of annotated examples used for labeling the test set on the accuracy, for MNIST (left) and Fashion-MNIST (right).

[3] Dey, L., Chakraborty, S., Nagwani, N.K.: Performance comparison of incremental k-means and incremental DBSCAN algorithms. International Journal of Computer Applications **27**(11), 14–18 (2011)

[4] Furao, S., Hasegawa, O.: An incremental network for on-line unsupervised classification and topology learning. Neural networks **19**(1), 90–106 (2006)

[5] Ge, Z., Demyanov, S., Chen, Z., Garnavi, R.: Generative openmax for multi-class open set classification. In: BMVC (2017)

[6] Geng, C., Huang, S.J., Chen, S.: Recent advances in open set recognition: A survey. IEEE transactions on pattern analysis and machine intelligence (2020)

[7] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS. pp. 2672–2680 (2014)

[8] Goyal, S., Benjamin, P.: Object recognition using deep neural networks: A survey. arXiv preprint arXiv:1412.3684 (2014)

[9] Joshi, P., Kulkarni, P.: Incremental learning: Areas and methods-a survey. International Journal of Data Mining & Knowledge Management Process **2**(5), 43 (2012)

[10] Kingma, D.P., Welling, M.: Auto-encoding variational bayes. stat **1050**, 1 (2014)

[11] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences **114**(13), 3521–3526 (2017)

[12] Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: NeurIPS. pp. 6402–6413 (2017)

[13] LeCun, Y.: The MNIST database of handwritten digits. http://yann.lecun. com/exdb/mnist/ (1998)

[14] Lesort, T., Lomonaco, V., Stoian, A., Maltoni, D., Filliat, D., Díaz-Rodríguez, N.: Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. Information fusion **58**, 52–68 (2020)

[15] Liang, S., Li, Y., Srikant, R.: Enhancing the reliability of out-of-distribution image detection in neural networks. In: ICLR (2018)

[16] Liu, W., Wang, X., Owens, J.D., Li, Y.: Energy-based out-of-distribution detection. arXiv preprint arXiv:2010.03759 (2020)

[17] Mallya, A., Davis, D., Lazebnik, S.: Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In: ECCV. pp. 67–82 (2018)

[18] Maltoni, D., Lomonaco, V.: Continuous learning in single-incremental-task scenarios. Neural Networks **116**, 56–73 (2019)

[19] Markou, M., Singh, S.: Novelty detection: a review—part 1: statistical approaches. Signal processing **83**(12), 2481–2497 (2003)

[20] Montiel, J., Read, J., Bifet, A., Abdessalem, T.: Scikit-multiflow: A multi-output streaming framework. Journal of Machine Learning Research **19**(72), 1–5 (2018), `http://jmlr.org/papers/v19/18-251.html`

[21] Page, E.S.: Continuous inspection schemes. Biometrika **41**(1/2), 100–115 (1954)

[22] Parisi, G.I., Kemker, R., Part, J.L., Kanan, C., Wermter, S.: Continual lifelong learning with neural networks: A review. Neural Networks **113**, 54–71 (2019)

[23] Rao, D., Visin, F., Rusu, A., Pascanu, R., Teh, Y.W., Hadsell, R.: Continual unsupervised representation learning. In: Advances in Neural Information Processing Systems. pp. 7645–7655 (2019)

[24] Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: CVPR. pp. 2001–2010 (2017)

[25] Ren, J., Liu, P.J., Fertig, E., Snoek, J., Poplin, R., DePristo, M.A., Dillon, J.V., Lakshminarayanan, B.: Likelihood ratios for out-of-distribution detection. In: NeurIPS (2019)

[26] Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. arXiv preprint arXiv:1606.04671 (2016)

[27] Sculley, D.: Web-scale k-means clustering. In: Proceedings of the 19th international conference on World wide web. pp. 1177–1178 (2010)

[28] Serrà, J., Alvarez, D., Gómez, V., Slizovskaia, O., Núnez, J.F., Luque, J.: Input complexity and out-of-distribution detection with likelihood-based generative models. In: ICLR (2020)

[29] Smith, J., Taylor, C., Baer, S., Dovrolis, C.: Unsupervised progressive learning and the STAM architecture. In: IJCAI (2021)

[30] Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-first AAAI conference on artificial intelligence (2017)

[31] Tolstikhin, I., Bousquet, O., Gelly, S., Schoelkopf, B.: Wasserstein auto-encoders. In: ICLR (2018)

[32] Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)

[33] Xu, R., Wunsch, D.: Survey of clustering algorithms. IEEE Transactions on neural networks **16**(3), 645–678 (2005)

[34] Zenke, F., Poole, B., Ganguli, S.: Continual learning through synaptic intelligence. In: International Conference on Machine Learning. pp. 3987–3995 (2017)

[35] Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: An efficient data clustering method for very large databases. In: Proc. of the ACM SIGMOD Intern. Conf. on Management of Data. p. 103–114 (1996)