# Self-supervised Novelty Detection for Continual Learning: A Gradient-based Approach Boosted by Binary Classification

**Jingbo Sun**[1] , **Yang Li**[1] , **Jiaxin Zhang**[2] , **Frank Liu**[2] , **Mahantesh Halappanavar**[3]
**Deliang Fan**[1] and **Yu Cao**[1]
[1]Arizona State University
[2]Oak Ridge National Laboratory
[3]Pacific Northwest National Laboratory
{jsun127, ycao}@asu.edu

## Abstract

Novelty detection aims to automatically identify out-of-distribution (OOD) data, without any prior knowledge of them. It is a critical step in continual learning, in order to sense the arrival of new data and initialize the learning process. Conventional methods of OOD detection perform multi-variate analysis on an ensemble of data or features, and usually resort to the supervision with OOD data to improve the accuracy. In reality, such supervision is impractical as one cannot anticipate the anomalous data. In this paper, we propose a novel, self-supervised approach that does not rely on any pre-defined OOD data: (1) The new method evaluates the Mahalanobis distance of the gradients between the in-distribution and OOD data. (2) It is assisted by a self-supervised binary classifier to guide the label selection to generate the gradients, and maximize the Mahalanobis distance. In the evaluation with multiple datasets, such as CIFAR-10, CIFAR-100, SVHN and ImageNet, the proposed approach consistently outperforms state-of-the-art supervised and unsupervised methods in the area under the receiver operating characteristic (AUROC). We further demonstrate that this detector is able to accurately learn one OOD class in continual learning.

## 1 Introduction

Deep neural networks (DNNs) have achieved high accuracy in many fields, such as image classification, natural language processing, and speech recognition. Their success is built upon carefully handcrafted DNN architectures, big data collection and expensive model training. A well-trained model promises high inference accuracy if the input falls into the distribution of the training data. However, in many real-world scenarios, there is no guarantee that the input is always in the distribution. The encounter with out-of-distribution (OOD) input is inevitable due to the difficulty in data collection, unforeseeable user scenarios, and complex dynamics.

To manage the emergence of OOD data at the first moment, it is vitally important to have an accurate novelty detector that continuously evaluates the data stream and alarms the system once OOD data arrives. Upon the detection of OOD arrival, the system can then manage the situation with three possible methods: (1) It can rely on the OOD detector to collect new data and send them back to the data center, such that OOD data can be combined with previous in-distribution data (IDD) to re-train the model; (2) It can temporally utilize the detector as a one-class classifier to recognize the new class of OOD, in addition to existing IDD classes; and (3) It can activate a continual learning method at the edge to adapt the model in the field, such as the multi-head method with the assistance of the OOD detector. In all three cases, the accuracy and robustness of the novelty detector guard the success of continual model adaptation and knowledge update.

In this work, we propose a self-supervised approach which generates a set of OOD data from unsupervised statistical analysis, instead of supervised labeling. This set of OOD data is then combined with IDD data to train a binary classifier, which in turn, helps boost the performance of the unsupervised detector that collects more OOD data for training. As this mutual process continues, our novelty detector achieves higher and higher confidence in OOD detection. The contributions of this paper are as follows:

- Gradient-based novelty detection that employs the Mahalanobis distance as the metric to differentiate in-distribution and OOD input. The gradients are generated from a pre-trained classifier for IDD only, without any pre-knowledge of OOD.
- A self-supervised binary classifier. As previous works demonstrated, the availability of a binary classifier helps boost the accuracy. Yet distinguished from them, we don't rely on any labelled OOD data to train the classifier. The training set is initialized by the gradient-based detector. In turn, the binary classifier pre-screens OOD and IDD, and guides the selection of labels in the gradient-based detector to maximize the distance calculation. Through such mutual assistance, our approach is unsupervised and continually improves the accuracy.
- High accuracy in OOD detection and one-class classification. We evaluate our methodology in a comprehensive set of benchmarks. As shown in Fig. 1, our self-supervised method consistently achieves higher AUROC than other supervised and unsupervised results, confirming the advantages in gradient-based novelty detection.

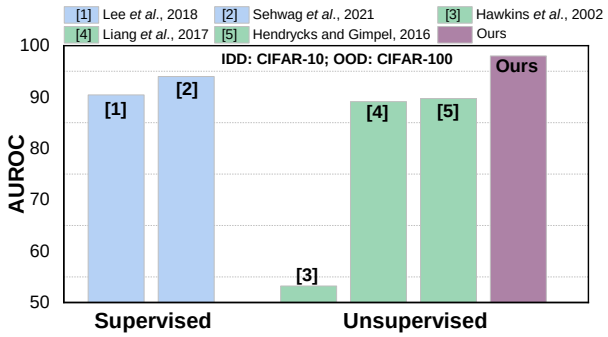


Figure 1: Comparison of AUROC between our proposed method and other state-of-the-art methods.

## 2 Previous Works on Novelty Detection

Current OOD detectors usually use the information extracted from either the data itself or features projected by the feedforward path in the IDD engine. [Hendrycks and Gimpel, 2016] demonstrated that the softmax score of the outlier tended to be higher compared with IDD and thus, thresholding inputs based on this score is feasible to detect the OOD. [Liang *et al.*, 2017] improved this idea by introducing input preprocessing and temperature scaling. [Lee *et al.*, 2018] proved that the in-distribution samples formed a multivariate Gaussian distribution in the high dimensional feature space, and proposed to use the Mahalanobis distance to measure how far an outlier is away from this in-distribution. There also exist autoencoder based OOD detectors [Abati *et al.*, 2019; Chen *et al.*, 2017; Hawkins *et al.*, 2002; Kwon *et al.*, 2020; Sakurada and Yairi, 2014; Zhou and Paffenroth, 2017] which use the reconstruction loss from the decoder to characterize the novelty. These data or activation-based methods demonstrated the value in OOD detection. On the other side, they have not explored one important step in the development of DNNs, the gradients back-propagated from the classification layer. These gradients present the first-order derivative to adapt the model and improve the separation of multiple classes. They contain a rich body of information to differentiate OOD from IDD.

To collect the gradients from the classifier that is prepared for the IDD, a label is required for cross-entropy loss and back-propagation. However, one challenge in the gradient-based approach is that labels of OOD data are not available in the process of novelty detection. To address this issue, a recent work by [Lee and AlRegib, 2020] introduced the confounding label, which only triggers small gradients for the IDD input. The gradients of the OOD input would be larger since they introduce many new features that are different from IDD. [Kwon *et al.*, 2020] explored the gradient-based representations of the novelty but they avoid the label issue by proposing a directional gradient constraint to the loss function so that the gradient direction of the OOD input does not align with the ones of the IDD. However, this method requires retraining of the model. In contrast, in this work, we utilize the gradient-based approach and the pre-trained model without any modification. In addition, we only use the training labels to collect the gradient.

Note that to boost the accuracy in novelty detection, many prior approaches utilize supervised training. They adopted a small amount of OOD samples to pre-train the novelty detector. For example, [Lee *et al.*, 2018] trained a logistic regression detector to estimate the weights for feature ensemble. [Lee and AlRegib, 2020] trained a binary detector to distinguish the gradient representation of the in-distribution and OOD input. However, this type of supervised training requires the availability of labelled OOD data up front, which restricts its application in reality.

## 3 Proposed Methodology

The overarching goal of our methodology is to accurately identify OOD data from IDD. A successful OOD detection is equivalent to correctly classify the OOD input as one new class (i.e., one-class classification). For IDD inputs, they will be classified to the previous known classes. To achieve this goal, we propose a closed-loop methodology that interleaves the unsupervised ODD detector based on the Mahalanobis distance, with a binary IDD/OOD classifier. Fig. 2 illustrates our methodology, consisting of these two main components:

- A gradient-based novelty detector: Distinguished from previous works that analyzed either the input data or the features, our detector exploits the gradients propagated backward from the classification layer for statistical analysis. The classifier in this step is designed for IDD only and all labels are from the IDD classes. To maximize the Mahalanobis distance between OOD and IDD classes, we select the appropriate class to label the OOD data, which is predicted by a binary classifier, as in Fig.2.
- A self-supervised binary classifier. This classifier is designed to pre-screen IDD and OOD data, in order to assist label selection to generate the gradients. While the structure of this binary classifier is similar as that in [Goodfellow *et al.*, 2014], the training does not rely on any labeled data, but from a balanced set that is selected by the detector based on the Mahalanobis distance.

Overall, the mutual assistance between these two components helps accomplish OOD detection without the need of external supervision. In the following subsections, we first introduce our main processing path in OOD detection: the Mahalanobis distance in the gradient space. We then introduce the binary classifier as a necessary assistant to boost the performance of the main path, as well as the self-supervised training of it. Finally, we describe how these two components mutually assist each other with a thorough case study.

### 3.1 Gradient-based novelty detector

Our approach starts from a given dataset of in-distribution data (IDD), and a deep neural network (DNN) based classifier trained by this IDD dataset. Note that at this step, only labels for IDD classes are accessible. In our examples of the image classification task, previous works have demonstrated that such a DNN is capable of separating the manifolds of each in-distribution class and achieve high classification accuracy. After the training of the DNN is completed, the gradients of each in-distribution sample, which is back-propagated from its ground-truth label, is distributed within a small range around

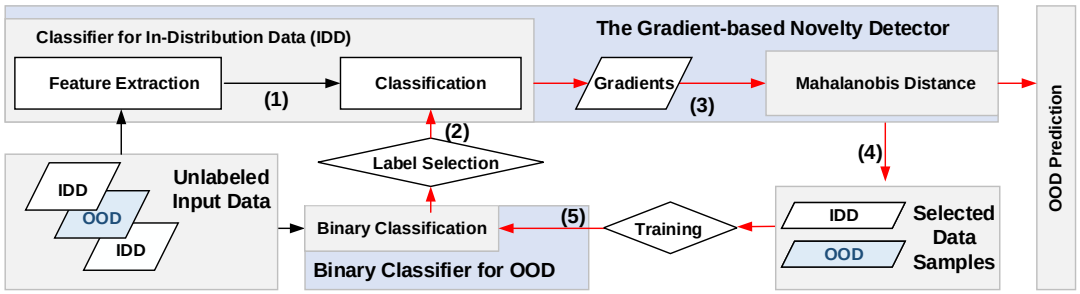


Figure 2: The flow of our self-supervised OOD detector. (1) A DNN-based classifier is first prepared for IDD data. (2) The binary classifier predicts IDD or OOD. If the result is OOD, then in the IDD classifier, it selects the appropriate label to maximize the distance. (3) Based on the selected label, gradients are generated by the IDD classifier for the calculation of Mahalanobis distance. (4) From each patch of data, a balanced set of IDD and OOD data is selected based on the Mahalanobis distance. (5) This balanced IDD/OOD data set is used to continuously train the binary classifier.

the manifold of its predicted class, forming class-specific distributions that correspond to each individual manifold.

When an OOD data is given to this pre-trained DNN, it will lie far away from any manifold of IDD. Without any knowledge about this OOD data, if we still perform gradient-based training supervised by the IDD label, the DNN model will experience much larger gradients which force the model to reconstruct itself toward the OOD data. In this context, the distribution of gradients will generate different characteristics for IDD and OOD, paving a promising path toward novelty detection. Therefore we propose to utilize the Mahalanobis distance in the gradient space as the statistical metric to separate the outliers from the in-distribution ones.

The Mahalanobis distance is defined by the following:

$$M_x = \min_c (\nabla_c f(x) - \hat{\mu}_c)^T \hat{\Sigma}^{-1} (\nabla_c f(x) - \hat{\mu}_c) \quad (1)$$

where $\hat{\mu}_c$ is the gradient mean of the in-distribution samples $\mathcal{X}_{in}$ of the class $c$ and $\hat{\Sigma}^{-1}$ is the tied precision matrix of all the known(training) classes. We use the equations below to estimate these two parameters:

$$\hat{\mu}_c = \frac{1}{N_c} \sum_{i:y_i=c} \nabla_c f(x_i^{in}) \quad (2)$$

$$\hat{\Sigma} = \frac{1}{N} \sum_c \sum_{i:y_i=c} (\nabla_c f(x_i^{in}) - \hat{\mu}_c)(\nabla_c f(x_i^{in}) - \hat{\mu}_c)^\top \quad (3)$$

where $N_c$ is the amount of the IDD samples in the class $c$, $N$ is the amount of the entire IDD training dataset.

We use Equations (2) and (3) to characterize the gradient distribution of the IDD by using the same training dataset of the DNN. After this class-specific distributions estimation is done, we use equation (1) to measure how the gradient of the new input deviates from these estimated distributions.

In Equation (1), $\nabla_c f(x)$ is the back-propagated gradient with respect to the class $c$. This raises a question of how to calculate the gradient of the OOD input since their ground-truth label $Y_{ood}$ is not in the IDD label space $\mathcal{Y}_{in}$. To solve this problem, we propose to use the predicted label $Y_{ood}^{Pred} \in \mathcal{Y}_{in}$ to calculate and cross-entropy loss $\mathcal{L}(Y_{ood}^{Pred}; X_{ood}; \Theta)$ and do the back-propagation. From the manifold perspective, this means that we select the manifold closest to the input sample. This manifold requires the minimal amount of adaptation to the new input, i.e., the minimal gradient and Mahalanobis distance. If such Mahalanobis distance is still large, the input sample has a high probability to be an outlier.

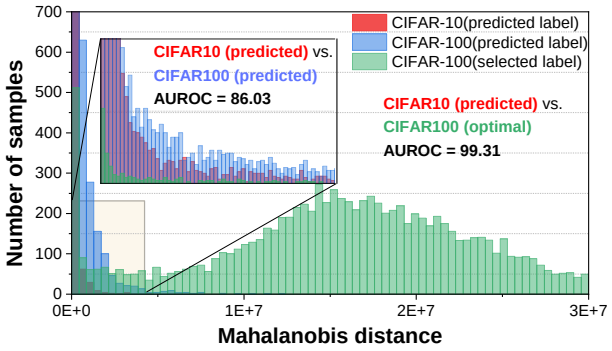


Figure 3: The distribution of the Mahalanobis distance of in-distribution data using predicted label (highlighted in red), and OOD using both predicted label (highlighted in blue) and selected label (highlighted in green). By using the selected label for all the outliers, their novelty score distribution shifts away farther away from blue to green, as shown in the main figure, with less overlap with IDD data (the red distribution, as shown in the inset.)

As shown in Table 1, if only using the predicted label by the IDD classifier, our proposed method is not competitive to the state-of-the-art method. This is due to the high overlap of the novelty score distribution between the IDD and OOD. In fact, the Mahalanobis distance is minimal if we use the predicted label as the ground truth label for back-propagation. To overcome this barrier, we intentionally select a different label to maximize the Mahalanobis distance for all the OOD samples. By doing that, we expect to achieve a larger mean of the novelty score distribution for the OOD. Consequently, it will be easier to threshold these OOD samples to reach higher detection accuracy.

Fig. 3 illustrates the novelty score distribution of the IDD using the predicted label and OOD using both predicted and a pre-selected label. After intentionally using the selected label for all the OOD samples, the corresponding score distributions

| IDD (CIFAR-10) | AUROC | | |
|---|---|---|---|
| OOD | State-of-the-art | Ours (with predicted label) | Ours (with selected label) |
| TinyImageNet | 99.50 | 91.10 | **99.92** |
| SVHN | 99.90 | 91.63 | **99.99** |
| LSUN | 99.70 | 90.08 | **99.99** |
| CIFAR-100 | 93.40 | 86.03 | **97.99** |

Table 1: Comparison of AUROC results between the state-of-the-art [Lee *et al.*, 2018; Sehwag *et al.*, 2021] and our proposed novelty detector. The middle column is the performance of our novelty detector using the predicted label from the DNN for IDD to calculate the loss, gradients and Mahalanobis distance. The right column shows the performance boosting if we intentionally select the label to maximize the distance between IDD and OOD.

shift away from the in-distribution and thus, significantly improves the AUROC result (Table 1, Column 3). However, this raises a new problem: how to trigger the usage of different labels for IDD and OOD (i.e., predicted by the IDD classifier and the selected label, respectively), in the calculation of gradients? Here we introduce a binary classifier to pre-screen the input and make the initial IDD/OOD prediction. Based on the prediction result, our novelty detector chooses different label for gradient calculation.

**Label selection:** To select a label $Y^{Opt}$ that gives the maximum gradient distance, we propose to use the predicted softmax class probability:

$$Y^{Opt} = \underset{c}{\operatorname{argmin}}(\sum_{i} Softmax(f(x_i^{ood}; \Theta))) \qquad (4)$$

We use Equation (4) to select the class that has the minimal average softmax probability for a batch of OOD data. This is equivalent to finding the least likely class for an OOD data to fall into. By using $Y^{Opt}$, its cross-entropy loss $\mathcal{L}(Y^{Opt}; X_{ood}; \Theta)$ becomes the maximum which results in the largest gradient and Mahalanobis distance. The only question left is where to find the OOD samples to estimate this label. We will address this issue in subsection 3.2.

## 3.2 Self-supervised binary classifier

To guide our novelty detector to use either the predicted or selected label in case of in-distribution or OOD input, we introduce a simple self-supervised binary classifier to screen IDD and OOD. The output from this binary classifier will be used to guide label selection in the pre-trained IDD classifier for gradient generation:

- For the predicted IDD input, the pre-trained model uses the predicted label $Y^{Pred}$ for back-propagation.
- For the predicted OOD input, the model uses the selected label $Y^{Opt}$.

**Unsupervised preparation of OOD samples for initial training:** One key feature of our proposed method is to be self-supervised, which means no OOD sample is available in advance. Therefore to create a dataset for the training of the binary classifier, we utilize the predicted IDD/OOD samples selected by the gradient-based novelty detector. For example, assuming the binary classifier is randomly initialized and a batch $X^1$ mixed with IDD and OOD data comes in, our novelty detector will first calculate the novelty confidence score $S^1 = \{s_1, s_2, ..., s_N\}_1$ using the predicted label. We select N/2 samples that correspond to the highest and lowest confidence scores in the $S^1$ as the binary classifier training data set $X^{pred} = \{X_{in}^{pred}, X_{ood}^{pred}\}$ . This step helps select the best possible in-distribution and OOD data from the current batch so that the binary classifier's training inputs are reasonable. We use the $X_{ood}^{pred}$ to select the label and use the $X^{pred}$ to train the binary classifier. Once training is done, the following input batches will involve the cooperation from both the gradient-based novelty detector and the binary classifier.

## 3.3 Mutual assistance between the binary classifier and the Mahalanobis path

The entire system is continuously exposed to a stream of unlabeled mini batch $X^1, X^2$,..., where each $X^i$ consists of N samples $\{x_1, x_2, ..., x_N\}$ mixed with IDD and OOD data. Due to the small size of the first batch $X^1$ and the relatively low performance of the novelty detector using the predicted label, the initial training of the binary classifier could not guarantee to be success. Therefore, an enhanced training (Fig. 2 and Algorithm 1) is required when more data is available. The training routine consists of three major steps: (1) Initial prediction of the binary classifier; (2) Calculation of the Mahalanobis score; and (3) Re-training of the binary classifier.

**Initial prediction of the binary classifier :** When the new batch $X^k = \{x_1, x_2, ..., x_N\}_k$ arrives, the system first concatenates this new batch $X^k$ with all the previously stored batches $X^{k-1}...X^1$. Given this concatenated batch, the binary classifier makes its initial outlier prediction $\{y'_1, y'_2, ..., y'_{N\times k} | y'_i \in (0, 1)\}$.

**Calculation of the Mahalanobis score:** The gradient-based novelty detector takes the prediction result from the binary classifier and the concatenated batch. For each sample $x_i$ in batch, our novelty detector first classifies it to one of the in-distribution class $y_i^{Pred} \in \mathcal{Y}_{in}$ , then checks the binary classifier prediction $y'_i$. If $y'_i$ is 0 (predicted in-distribution), use $y_i^{Pred}$ as the ground truth label in the loss function for back-propagation, otherwise, use the pre-selected label $y^{Opt}$. After the gradient is available, we calculate the Mahalanobis distance $s_i$ as the novelty confident score.

**Re-training of the binary classifier:** Given novelty confident score $S^k = \{s_1, s_2, ..., s_{N\times k}\}$ from the gradient-based novelty detector, we select $(N \times k)/2$ samples from the concatenated batch that correspond to the highest and lowest scores in $S^k$ as a new training dataset for the binary classifier. Before training, we re-initialize the binary classifier to make sure the previous model will not be inherited into the current stage. Once training is done, our system is updated with the knowledge of all previous batches and is ready to process next available inputs with higher detection accuracy.

As shown in Fig. 2, this mutual assistance continues with more unlabelled data, which keeps improving the accuracy of both the unsupervised detection engine and the binary classifier. These two units help each other in this closed loop.

**Algorithm 1** Gradient-based Novelty Detection Boosted by Self-supervised Binary Classification

**Input:** In-distribution gradient distributions $\{\hat{\mu}, \hat{\Sigma}^{-1}\}$, batches for testing $[X^1, X^2, ..., X^k]$

**function** $noveltyScore\ (X, \hat{\mu}, \hat{\Sigma}^{-1}, \hat{Y}, sel\_label)$
  $Mahalanobis\_Distance = [\,]$
  **for each** $x\ in\ X$ **do**
    $c = None\quad //Label\ for\ back\text{-}propagation$
    **if** Binary Classifier predict x as IDD **then**
      $c$ = Novelty detector predicted label
    **else**
      $c = sel\_label$
    **end if**
    $Score = (\nabla_c f(x) - \hat{\mu}_c)^T \hat{\Sigma}^{-1} (\nabla_c f(x) - \hat{\mu}_c)$
    $Mahalanobis\_Distance.append(Score)$
  **end for**
  **return** *Mahalanobis_Distance*
**end function**
$X^{all} = None$
**while** new batch $X^k$ is available **do**
  $X^{all} = X^{all} + X^k \quad //batch\ concatenation$
  **if** $X^k$ is the first batch **then**
    $S^{all} = novelty\_score(X^{all}, \hat{\mu}, \hat{\Sigma}^{-1})$
  **else**
    $\hat{Y} = binary\ classifier\ prediction\ on\ X^{all}$
    $S^{all} = noveltyScore(X^{all}, \hat{\mu}, \hat{\Sigma}^{-1}, \hat{Y}, sel\_label)$
  **end if**
  Based on $S^{all}$, select samples from $X^{all}$
with the highest/lowest score as predicted $X_{in}/X_{ood}$
  Use $X_{in}, X_{ood}$ to train binary classifier
  **if** $sel\_label\ is\ None$ **then**
    $sel\_label = \underset{c}{\operatorname{argmin}}(\sum_i Softmax(f(x_i^{ood})))$
  **end if**
**end while**
**return** $X_i \in X_{all}$ is an outliner if $S_i^{all} > threshold$

# 4 Experiments and Results

## 4.1 Experimental setup

We use three pre-trained ResNet-34 networks [He *et al.*, 2016] provided by [Lee *et al.*, 2018] as the base of our gradient-based novelty detector. Each model is trained on CIFAR-10, CIFAR-100 [Krizhevsky *et al.*, 2009] and SVHN [Netzer *et al.*, 2011] with the testing accuracy of 93.67%, 78.34% and 96.68% accordingly. To calculate the gradient-based Mahalanobis distance, we only use the gradient extracted from the last layer of the feature extractor. Our simple binary classifier has the structure of three convolution layers and one batch normalization layer with a Sigmoid classifier. It is trained by minimizing the cross-entropy loss using Adam [Kingma and Ba, 2014]. The initial learning rate is set to 0.0002 and the decay rate is controlled by $\beta_1 = 0.5$ and $\beta_2 = 0.999$. We train it for 500 epochs in both initial training and re-training process. Regarding the batch size, we find that the amount of IDD and OOD samples in each batch, and the batch size itself have a strong impact on the performance of the binary classifier and the overall system, as discussed in subsection 4.2.

Our experiment includes three in-distribution datasets: CIFAR-10, CIFAR-100 and SVHN. We test each of them using the other two as the OOD. In addition, we use two more OOD datasets: the resized version of the ImageNet [Russakovsky *et al.*, 2015] and LSUN [Yu *et al.*, 2016] provided by [Liang *et al.*, 2017]. Each of these two datasets contains 10,000 images with size 32 by 32. For all three in-distribution datasets, we use only the testing portion of the images because the training portion has already been used to train the novelty detector. For each IDD/OOD pair, we randomly select 5,000/5,000 (IDD/OOD) as the training dataset and further divide them into mini batches with size of 100/100 (IDD/OOD) to emulate the data streaming, the rest 5,000/5,000 (IDD/OOD) samples are used to test the binary classifier accuracy and the overall novelty detection performance after each new batch has been taken into the system. We evaluate our detector with two performance metrics: AUROC and AUPR. All experiments are performed with PyTorch [Paszke *et al.*, 2017] on one NVIDIA GeForce RTX 2080 platform.

## 4.2 Batch training of the binary classifier

Inspired from the discriminator in the GAN [Goodfellow *et al.*, 2014] that can successfully distinguish the real images from the fake ones, we use the similar discriminator loss in our binary classifier training, where the loss from the IDD and OOD data are calculated separately. During the prediction phase, the binary classifier achieves high accuracy if the input batch contains either in-distribution or OOD samples. This is because the batch norm layer averages the difference within each group of data (i.e., IDD or OOD) while stressing the difference between IDD and OOD inputs. For each novelty detection experiment, we adopt this batch-based approach and present the results of using different batch size where each batch contains only IDD or OOD data.

## 4.3 Performance evaluation of novelty detection

Table 2 presents the performance of our proposed method as compared to other novelty detectors. Our method outperforms all previous supervised and unsupervised works across all IDD/OOD setup. In particular, our method improves the AUROC of the experiment where CIFAR100 as IDD and CIFAR10 as OOD by up to $\sim$5 and $\sim$7 with batch size 32 and 128 accordingly. These two datasets are very similar to each other. Therefore it's extremely challenging to detect the outlier in previous approaches. With the new framework, our method significantly improves the state-of-the-art.

Fig. 4 illustrates the stepwise improvement of our framework. Each point in the curve corresponds to the intermediate testing accuracy of the binary classifier and the AUROC of the system after every new batch of unlabelled data (100/100) arrives. From the curves, we can observe the efficacy of mutual assistance between the binary classifier and the novelty detector. The initial testing accuracy of the binary classifier reaches around 90 in all three IDD/OOD experiments, which proves that our self-supervised approach based on the novelty detector's output is effective. As more data is received, every re-training on the binary classifier improves its accuracy, contributed by increasingly higher confidence of the IDD/OOD prediction from the novelty detector. In turn, the novelty

| Dataset | | AUROC | | | | | | |
|---|---|---|---|---|---|---|---|---|
| IDD | OOD | Ours (Batch=8) | Ours (Batch=32) | Ours (Batch=128) | SSD[1] | Mahalanobis[2] (feature based) | Confounding label[3] | ODIN[4] |
| CIFAR-10 | TinyImageNet | 99.90 | **99.92** | 99.85 | - | 99.5 | 93.18 | 98.5 |
| | CIFAR-100 | 84.79 | 93.51 | **97.99** | 94.0 | - | - | - |
| | LSUN | **99.99** | 99.97 | **99.99** | - | 99.7 | 99.86 | 99.2 |
| | SVHN | 99.94 | **99.99** | **99.99** | 99.9 | 99.1 | 99.84 | - |
| CIFAR-100 | TinyImageNet | **99.49** | 99.28 | 99.28 | - | 98.2 | - | 85.5 |
| | CIFAR-10 | 72.61 | 89.76 | **91.38** | 84.0 | - | - | - |
| | LSUN | 99.57 | **99.59** | 99.55 | - | 98.2 | - | 86.0 |
| | SVHN | 99.79 | 99.79 | **99.82** | 99.5 | 98.4 | - | - |
| SVHN | CIFAR-10 | 99.94 | **99.96** | 99.95 | - | 99.3 | 99.79 | - |
| | TinyImageNet | 99.95 | **99.97** | 99.95 | - | 99.9 | 99.77 | - |
| | LSUN | 99.96 | 99.97 | **99.98** | - | 99.9 | 99.93 | - |
| | CIFAR-100 | 99.65 | 99.74 | **99.78** | - | - | - | - |

[1] [Sehwag *et al.*, 2021].[2] [Lee *et al.*, 2018].[3] [Lee and AlRegib, 2020].[4] [Liang *et al.*, 2017]

Table 2: Comprehensive evaluation of AUROC on multiple IDD and OOD benchmarks.

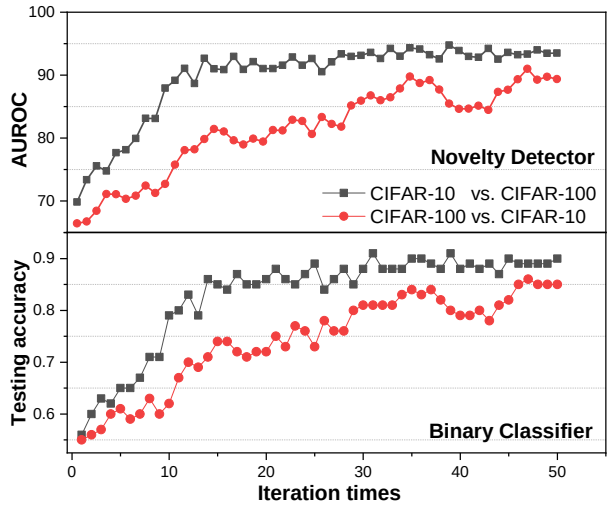


Figure 4: AUROC in our novelty detector (top) and the testing accuracy of the binary classifier (bottom) during the mutual assistance (i.e., number of iterations). Each point corresponds to the moment when a new batch arrives into the system. As the accuracy of the binary classifier increases, the novelty detector receives more boosting.

detector's performance is boosted, benefiting from higher accuracy of the binary classifier and better label selection. Such positive feedback eventually drives the performance of the overall framework, reaching high accuracy of both the novelty detector and the binary classifier.

### 4.4 One-class learning with the OOD detector

We further apply our OOD detector to one-class learning, using CIFAR-10 as the example. In this case, we first train the ResNet-34 network with 9 classes (IDD) together and then consider the last tenth class as the new learning task (OOD). Different from previous experimental setup where an equal amount of IDD/OOD samples (5,000/5,000) are used, we attempt to minimize the amount of IDD samples (i.e., 500 and 1,000 in our experiments) to emulate the memory rehearsal. We still allow more OOD samples (2,500) streaming into the system. In addition to predict each testing input as either IDD or OOD, we further send all the predicted IDD samples back to the pre-trained IDD classifier to recognize the belonging to those 9 IDD classes. Therefore, this procedure completes one-class continual learning. In the field, this solution will temporally help the system to manage incoming OOD data until a new model with updated IDD+OOD classes is available.

| CIFAR-10 (9+1) | Single-head Accuracy | |
|---|---|---|
| New task (OOD) | Memory budget=500 | Memory budget=1000 |
| Airplane | 90.58 | **95.56** |
| Automobile | 82.16 | **95.33** |
| Bird | 90.91 | **91.08** |
| Cat | 91.70 | **94.54** |
| Deer | **93.32** | 92.40 |
| Dog | 83.28 | **94.98** |
| Frog | **96.32** | 96.06 |
| Horse | 86.42 | **93.50** |
| Ship | **96.48** | 96.44 |
| Truck | 84.29 | **95.39** |

Table 3: Single-head accuracy tested on each class of CIFAR-10 in continual learning. 9+1: 9 classes are first learned together and the last one class is learned through the OOD detector.

We test the 9+1 single-head accuracy on each class with different memory budgets (Table 3). Compared with [Du *et al.*, 2020] which used memory budget of 2,000 IDD samples with the accuracy <90%, our proposed method achieves 91-96% accuracy in any class with only 1,000 IDD samples and competitive performance even with 500 IDD samples.

## 5 Conclusion

In this paper, we propose a new framework for novelty detection and one-class continual learning, which involves the cooperation of a gradient-based novelty detector and a self-supervised binary classifier. Our proposed framework outperforms previous supervised and unsupervised novelty detectors, as well as one-class continual learning. The success of the new approach promises accurate and robust novelty detection in continual learning and other applications.

## Acknowledgements

This work was supported by the U.S. Department of Energy's Office of Advanced Scientific Computing Research (DOE ASCR). It is also supported in part by C-BRIC, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA.

## References

[Abati *et al.*, 2019] Davide Abati, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Latent space autoregression for novelty detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 481–490, 2019.

[Chen *et al.*, 2017] Jinghui Chen, Saket Sathe, Charu Aggarwal, and Deepak Turaga. Outlier detection with autoencoder ensembles. In *Proceedings of the 2017 SIAM international conference on data mining*, pages 90–98. SIAM, 2017.

[Du *et al.*, 2020] Xiaocong Du, Zheng Li, Jae-sun Seo, Frank Liu, and Yu Cao. Noise-based selection of robust inherited model for accurate continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 244–245, 2020.

[Goodfellow *et al.*, 2014] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[Hawkins *et al.*, 2002] Simon Hawkins, Hongxing He, Graham Williams, and Rohan Baxter. Outlier detection using replicator neural networks. In *International Conference on Data Warehousing and Knowledge Discovery*, pages 170–180. Springer, 2002.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Hendrycks and Gimpel, 2016] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[Kwon *et al.*, 2020] Gukyeong Kwon, Mohit Prabhushankar, Dogancan Temel, and Ghassan AlRegib. Novelty detection through model-based characterization of neural networks. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 3179–3183. IEEE, 2020.

[Lee and AlRegib, 2020] Jinsol Lee and Ghassan AlRegib. Gradients as a measure of uncertainty in neural networks. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 2416–2420. IEEE, 2020.

[Lee *et al.*, 2018] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *arXiv preprint arXiv:1807.03888*, 2018.

[Liang *et al.*, 2017] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

[Netzer *et al.*, 2011] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[Paszke *et al.*, 2017] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[Russakovsky *et al.*, 2015] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[Sakurada and Yairi, 2014] Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, pages 4–11, 2014.

[Sehwag *et al.*, 2021] Vikash Sehwag, Mung Chiang, and Prateek Mittal. Ssd: A unified framework for self-supervised outlier detection. *arXiv preprint arXiv:2103.12051*, 2021.

[Yu *et al.*, 2016] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop, 2016.

[Zhou and Paffenroth, 2017] Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 665–674, 2017.