

Where's Wally Now?

Deep Generative and Discriminative Embeddings for Novelty Detection

Philippe Burlina, Neil Joshi, and I-Jeng Wang
 Johns Hopkins University Applied Physics Laboratory
 Fphi l i ppe. burl i na, nei l . joshi , i -j eng. wang@j huapl . edu

Abstract

We develop a framework for novelty detection (ND) methods relying on deep embeddings, either discriminative or generative, and also propose a novel framework for assessing their performance. While much progress was made recently in these approaches, it has been accompanied by certain limitations: most methods were tested on relatively simple problems (low resolution images / small number of classes) or involved non-public data; comparative performance has often proven inconclusive because of lacking statistical significance; and evaluation has generally been done on non-canonical problem sets of differing complexity, making apples-to-apples comparative performance evaluation difficult. This has led to a relative confusing state of affairs. We address these challenges via the following contributions: We make a proposal for a novel framework to measure the performance of novelty detection methods using a trade-space demonstrating performance (measured by ROCAUC) as a function of problem complexity. We also make several proposals to formally characterize problem complexity. We conduct experiments with problems of higher complexity (higher image resolution / number of classes). To this end we design several canonical datasets built from CIFAR-10 and ImageNet (IN-125) which we make available to perform future benchmarks for novelty detection as well as other related tasks including semantic zero/adaptive shot and unsupervised learning. Finally, we demonstrate, as one of the methods in our ND framework, a generative novelty detection method whose performance exceeds that of all recent best-in-class generative ND methods.

1. Motivation, prior work, and contributions

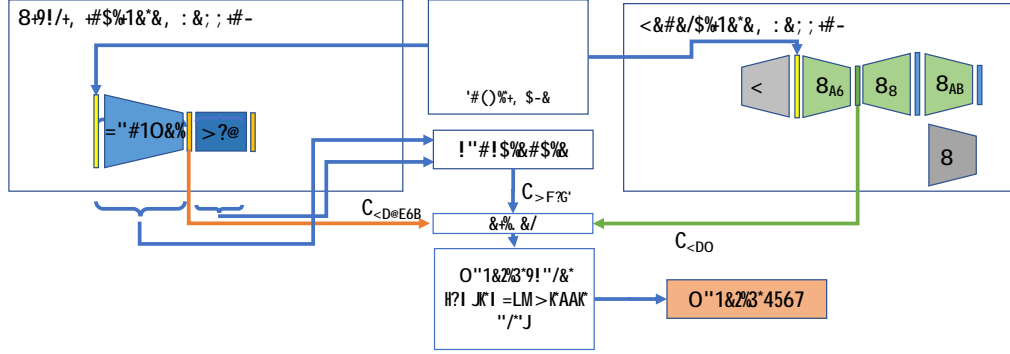
Motivation Novelty detection (ND) methods [7, 13–15, 18, 31, 33, 36, 38] have applications in a wide array of use cases including in semi/unsupervised learning, lifelong learning and zero-shot learning [29]. Application examples include: (1) robotic applications with lifelong and open set

learning abilities, where the ability to perform novelty detection allows a robot to trigger human-machine dialogue to seek information on novel objects it encounters; (2) performing medical diagnostics for rare diseases (e.g., myopathies) – for which prior observations are sparse or unavailable – and using novelty detection to pre-screen patients and refer them to clinicians; and (3), zero-shot semantic learning leveraging novelty detection to improve zero-shot classification performance by turning a *generalized / uninformed* zero-shot problem into an easier *informed* problem (where it is known if the object comes or not from a novel class) for which higher performance can be achieved [29, 38].

Prior work Past work in ND goes far and wide, and started with methods primarily grounded on classical machine learning (for recent surveys see [14, 31, 33], and more domain specific surveys [2, 4, 25]), including approaches such as [7, 13, 15]. For instance, [7] used image features and Support Vector Data Description (SVDD) for ND applied to hyperspectral imaging. [13] used convolutional sparse models to characterize novelty. Novel multiscale density estimators working for high dimensional data were developed and applied to ND in dynamic data in [15]. By contrast, representation learning via deep learning (DL) has offered novel ways to implement ND [18, 25, 36], with methods falling into two main categories: generative and discriminative.

Recent ND via DL Novelty detection using discriminative deep learning approaches were developed first based on deep embeddings computed by processing the image through deep discriminative networks, including deep convolutional neural nets (DCNNs), deep belief networks (DBNs), or recursive neural nets (RNNs). In [18] discriminative embeddings via DBNs and DCNNs were used along with one class SVM (1CSVM) to detect novelty. Most recently, generative approaches, principally via generative adversarial networks (GANs) or variational autoencoders (VAEs) have been embraced for representation learning for novelty detection [1, 3, 8, 16, 20, 21, 24, 26, 28, 32, 36, 41]. Most approaches focus on GANs’ ability to offer varied means for embedding (e.g., in latent space, using the en-

Figure 1. ND framework We propose a novelty detection (ND) framework along with a principled approach for evaluating ND methods which computes performance as $(\text{ROCAUC})=f(\text{problem complexity})$. Our ND framework performs novelty detection in two main steps, by a) embedding the image either via generative or discriminative networks, then b) computing a novelty score. Specifically: the left block shows our use of two types of discriminative embeddings (here via VGG16) using either GAP features $X_{\text{GAP}512}$ computed from the output of the convolutional layers (marked as ConvNet) or the concatenation of all features from the ConvNet layers as well as the fully connected layers (marked as MLP), producing an embedding vector denoted as X_{MULTI} . An alternative in our framework consists of using a generative embedding: using the latent space vector of a GAN (here we use Ganomaly [3]), denoted as X_{GAN} . Embedded feature vectors are then used to compute a novelty score via one of several approaches: either one class SVM (1CSVM), local outlier factor (LOF), elliptic envelope (EE) or isolation forest (IF). The Ganomaly architecture uses as a discriminator a series of encoder (D_{E1}) decoder (D_D) and encoder (D_{E2}). In our framework we directly leverage the latent vector X_{GAN} output of D_{E1} as embedding for ND.



coder or decoder network for embeddings, or using the discriminator's output, or other more complex ways) where novelty scores can then be computed. Looking at the core contributions: AnoGAN in [36] used DCGAN [34], and compared several GAN embedding and novelty measure approaches that included inverting the mapping from image to latent space, and measuring reconstruction error. It demonstrated improvements in area under the curve (AUC) when compared to simply using the GAN discriminator for novelty detection. It did so in experiments performing pixel-based anomaly detection on OCT images, using 49 OCT images and 64×64 sliding windows for pixel ND. A related method, ADGAN, was applied to whole image anomaly detection [16] and tested on images including MNIST (28×28 images) and CIFAR-10 (32×32)), showing some improvement over past GAN-based methods. [41] proposed a related approach using a more efficient GAN implementation and a modified loss function, and tested it on network intrusion data and a simple image dataset. Recently, [3] developed an approach using GANs where the discriminator network structure was composed of an encoder/decoder/encoder path, providing two latent embedded spaces, and the novelty score was measured via the reconstruction error between two latent spaces. This resulted in best in class performance when compared to all aforementioned methods on CIFAR-10 and MNIST.

Challenges Given the explosion of methods relying on deep embeddings for ND, one would hope to bring some order and address challenges in interpreting what constitutes actual progress in performance and what led to performance improvement. Some challenges stem from the difficulty in

making apples-to-apples comparisons due to the lack of repeated testing protocols. For instance, even when the same dataset is used, say MNIST, choosing different schemes for partitioning inlier and outlier classes results in large differences in difficulty for the resulting novelty detection problems. Quantitatively measuring ND problem complexity has never before been addressed, and is itself a complex challenge. Finally, most past studies have used rather simple problems with small number of classes and low resolution images, making it hard to predict how these methods would generalize in more complex, in-the-wild situations.

Contributions of this work

We address ND, defined as the problem of training on inlier data not corrupted by outliers, and making inferences on new observations to detect outliers. To address the ND challenge we make the following contributions: using a simple taxonomy of methods, we develop a framework for ND methods including both discriminative and generative embeddings, coupled with various approaches for measuring novelty. We make a proposal for a principled method for evaluating the performance trade space of ND methods, that expresses computed performance (AUC) as a function of measured problem complexity. We propose and discuss several methods for quantitatively assessing problem complexity based on semantic, information theoretic, and Bayes error based approaches. We propose, release, and test on canonical datasets and protocols for ND assessment, based on CIFAR-10 and ImageNet with higher image resolution and number of classes. Finally, we demonstrate that one of our ND generative methods exceeds performance when compared to all prior generative methods reported thus far.

2. Methods

We describe next the main approaches used herein in our ND framework using generative or discriminative deep embeddings. The family of algorithms we consider uses the following pipeline: a) computation of deep embedding, via discriminative methods, or using GANs for generative methods (Section 2.1) leading to an embedded vector X of an input image I ; b) PCA dimensionality reduction and normalization of X , and then followed by c) measuring novelty scores (Section 2.2). As novelty detection is trained on a set of inlier classes only, and testing is carried out on a set of images from inliers and outlier (yet unseen) classes, step (c) broadly consists of using training exemplars to characterize some notion of “distance” computed from a test image to training inlier exemplars, which is then translated into a novelty score. This framework and its various subcomponents are illustrated in Figure 1.

2.1. Discriminative and generative embeddings for ND

Discriminative embeddings In this work we start by computing deep embedding on the image using a pre-trained DCNN, here using VGG-16 [35, 37]. The structure of VGG-16 is recalled here only for convenience: it is first composed of a series of convolutional blocks:

$$C_{(2,64)}^1 \quad C_{(2,128)}^2 \quad C_{(3,256)}^3 \quad C_{(3,512)}^4 \quad C_{(3,512)}^5 \quad F$$

where a block named $C_{(n_l, n_d)}^i$ denotes the i th block composed of n_l successive convolutional layers of size 3×3 with depth n_d , each of which is followed by rectified linear units (ReLU) activation, and where each such block is followed by a pooling layer. This is then followed by flattening F and then fed to three successive fully connected layers on a vector of width 4096:

$$FC_{4096}^1 \quad FC_{4096}^2 \quad FC_{4096}^3$$

Our embeddings consist of computing both GAP (global average pooling) features, denoted X_{GAP512} , and multi-layer (X_{MULTI}) features. The GAP features are computed out of the output of $C_{(3,512)}^5$ and we apply the average operator to each of the 512 feature planes resulting in a 512-long feature vector. X_{GAP512} feature embedding forms a representation of input images that can be interpreted to contain low and mid level semantic information. In addition we also use as alternate embedding the concatenated feature outputs out of all layers (convolutional and fully-connected), denoted as X_{MULTI} , of total dimension 9664. Feature computation is followed by dimensionality reduction using PCA (with dimension equal to 120). Since this approach uses a pre-trained network, note that in experiments reported later, care is taken that no class used for pre-training the DCNN in ImageNet is used also as an outlier class to ensure that outlier

classes truly are unseen by the model. To take a stronger stand we also pose the same restrictions on inlier classes.

Generative embeddings Generative adversarial networks (GANs) are a deep generative approach which learns to generate novel images from a training dataset. GANs are composed of two networks that work with adversarial losses. One network performs image generation, using for example up-convolutions, that map a randomly sampled vector from latent space to image space, thereby generating synthetic images. Synthetic images thus created are then fed to a discriminative network (along with real images), and this network is trained to classify generated vs real images. The discriminative network may use a cross-entropy loss function or Jensen-Shannon divergence which it minimizes, while the generative network – tries to fool the discriminative network, and to maximize this loss function. At convergence, the discriminator has learned to discriminate between real and fake images, while the generator has learned to generate realistic looking images that are essentially sampled from the original training image distribution.

One possibility for a GAN embedding used for novelty detection consists of using the latent space. Computing it could be done by a network that performs an inversion of the generative mapping (an encoder network) that takes an image as input and generates a latent vector as output. As an alternative method, this latent vector can be fed back again through the GAN’s generator network, in essence creating a reconstruction of the original input image. One can then send this reconstruction through the GAN’s discriminator network to perform novelty detection.

An alternate method to obtain the latent vector is via the method in [3] which relied on a discriminator that used a decoder/encoder/decoder structure. Because of this structure, by training the discriminator, one essentially obtains an encoder that yields a latent space mapping “for free” (without inversion needed). Unlike [3], we use this approach in our framework as a means of producing a generative latent space embedding for novelty detection. We call this vector embedding X_{GAN} . In our approach, we use X_{GAN} via the ND scoring methods described in the next section.

Finally, we compare our ND generative approach to three methods: 1) the original Ganomaly ND scheme in [3], which relies on computing the latent vector reconstruction error between the output of the first and second encoder in the discriminator (See Figure 1). 2) Alternatively, the image reconstruction error can be used as a score of novelty, as is considered in [36, 41]. [41] in particular employed a novelty score based on a modification of the AnoGAN in [36]. We use it here also for performance comparisons and denote it as ND-EGAN [41] (for “efficient” GAN). 3) Another principle for GAN-based ND is based on the hypothesis that the discriminator may be used to detect anomalies.

We use a variation of this method exploiting the discriminator of Wasserstein GAN (WGAN) [5] which we denote “ND-DGAN”.

2.2. Novelty scores

Novelty detection is done by computing novelty scored on the aforementioned X_{GAP512} , X_{MULTI} , and X_{GAN} embedding vectors. We use four approaches which essentially measure the departure of a test vector X compared to a set of inlier samples, used for training. These are briefly reviewed below (for more details see the appendix):

LOF The first approach uses **local outlier factor** (or “LOF”) [9]. LOF computes a novelty score based on assessing the local density of points around a test point when compared to the density measured for each of this point’s neighbours. Intuitively, if a test point’s density is lesser than its neighbours’ densities, the point may be an outlier.

1CSVM The second approach here uses a **one class support vector machine** (or “1CSVM”) [7, 39]. 1CSVM is a large-margin non-parametric classifier that essentially proceeds by bounding a set of training inlier exemplars via the smallest enclosing hypersphere. In loose terms (see appendix for precise details) the novelty score is then a distance from the test vector to the weighted centroid of the support vectors on this hypersphere.

IF Isolation forest (later termed “IF”) is another novelty detection method used here [27]. It is akin to random forests in that it consists of building random CART tree structures on the training data and exploiting the fact that outliers typically stand isolated in a branch close to the root of the tree. Random features are selected and splits are computed on these features. The path from the root to a feature vector averaged over a set of random trees is taken as isolation score and therefore (short meaning isolated) used as novelty score.

EE Finally, we also use an ND score that takes the **elliptic envelope** (later termed “EE”) obtained by assuming a Gaussian distribution fit to the inlier training exemplars.

In subsequent experiments, for nomenclature: we denote the complete method applying a ND method “XXX” on feature type “YY” by “XXX/YY” (e.g., EE applied to GAP512 is denoted “EE/ X_{GAP512} ”).

2.3. Characterizing ND problem complexity

For comparing the different deep learning novelty detection approaches here, we use a set of canonical test problems mixing inlier and outlier classes: we thereby obtain a wide range of problems of varying complexity. We endeavor to characterize this complexity via some quantitative measure, to allow for an apples-to-apples comparison of problems of similar complexity. Our final goal is to assess how well the proposed ND methods perform for different complexity regimes. Intuitively, the complexity, and the

resulting ND performance, should depend on how *close* the inlier and outlier are in *distribution* or *semantics*. Here we have different choices for *closeness*:

One possible approach to complexity is to assess how semantically related the outlier classes are to inlier classes. This could be achieved by looking at proximity in a hierarchical class representation (e.g. ImageNet WordNet representation). However, one challenge with this approach is that semantic closeness would depend on the specific hierarchical representation utilized.

As an alternative, we opt for complexity assessment by characterizing closeness in distribution, here to be solved by evaluating distances of probability density of embedded vectors lying in high dimensional spaces. However, this endeavor is itself still an open research problem. Two approaches could be considered:

KLCA: KL-divergence based Complexity Assessment A first possibility is to characterize the distance of distributions (inlier and outlier) via information theoretic measures, to characterize ND complexity, e.g. using cross-entropy, earth moving distance or KL divergence. The computation of these measures in general, and KL divergence in particular, in high dimensions, is still an open problem for which recently some approaches have been proposed. We used in particular the method in [40] which leverages efficient K nearest neighbour (KNN) to compute KL. We call this method KLCA (for KL-based complexity assessment).

BERCA: Bayes error rate complexity assessment We propose a second approach for characterizing complexity which consists of assessing the complexity metric as the Bayes error rate of the two class problem associated with the one class problem partition. We denote this via the acronym “BERCA”. Following this route, we consider the two class classification problem Bayes decision rule, i.e. the rule that minimizes the probability of error P_e . We compute the empirical estimate of this error via KNN. Denoting by k_{knn} the numbers of neighbours used in KNN and n_{knn} the numbers of samples used, we recall [17] that the knn error rate P_e is such that $P_e = P_e^{\text{inlier}} + P_e^{\text{outlier}}$, where $P_e^{\text{inlier}} = 0$ when $k_{\text{knn}} = 10$ and $n_{\text{knn}} = 1000$. Henceforth, we use the Bayes error rate computed from the empirical KNN error rate estimate as the problem complexity metric. We use numerical values of $k_{\text{knn}} = 10$ and $n_{\text{knn}} = 1000$.

Standardizing the performance evaluation

Finally, equipped with the complexity measures above, we propose that for apples-to-apples comparisons, ND algorithmic performance be characterized in a trade space that expresses computed performance, measured in terms of ROCAUC, as a function of complexity of the ND problem where this performance was evaluated (i.e. $\text{AUC} = f(\text{complexity})$). This is to be displayed as box plots since the complexity of problems will vary, as will also the resulting performance range for a given complexity bin.

Figure 2. Examples of images for one of the problems run in IN-125. In this example 9 inlier classes are randomly selected from the IN-125 dataset (including bikes, bridges, etc.), and one outlier class (airplane) is also selected. We show examples of correct and incorrect novelty detection. For this experiment AUC=0.896

case, we perform experiments on SIMO and MISO problems defined above for $n_c = 10$ classes. In each problem case, we run 10 experiments, circularly rotating the single inlier (SIMO) or single outlier (MISO) every time. We then evaluate the AUC performance averaged across all 10 experiments (shown in Table 1).

Figure 3. Examples of images for one of the problems run in IN-125. In this example 9 inlier classes are randomly selected from the IN-125 dataset (including bunnies, carrots, cows, etc.), and one outlier class (Kangaroo) is also selected. We show examples of correct and incorrect novelty detection. Most incorrect classifications result from confusing factors such as multiple objects and humans, or bunnies/kangaroo similarity in appearance. For this experiment AUC=0.878

3. Standardizing ND problem structure and canonical datasets

We also propose to standardize the data used for evaluating ND around two main datasets with low and high resolution images and two main protocols for choosing inlier vs outlier class partitioning, described next.

MISO and SIMO problems In general, ND methods operate on the assumption that they are trained on inliers only, while testing is carried out on a set of inliers and outlier-class items, an assumption which is deemed to be semi-supervised. In addition, partitions of inliers and outliers sets can include either single or multiple classes. With that in mind, we consider two types of problems, and we adopt the following nomenclature: Single (class) Inlier, Multiple (class) Outlier problems (termed “SIMO” henceforth), and Multiple (class) Inlier, Single (class) Outlier problems (called “MISO”).

Recalling our first use case of an exploring robot as an illustrative example, and assuming the existence of n_c classes in the *universe*, SIMO problems can be thought of as one-vs-all problems, in which the robotic agent bootstraps its knowledge of the universe with a single starter inlier class, and we perform experiments that evaluates on that same inlier class, plus $n - 1$ other outlier classes that the agent may encounter, while doing exploration. MISO problems can be seen as leave-one-out problems, where the ND method is trained on $n_c - 1$ inlier classes, and tested on that same set of inlier classes, plus the remaining outlier class.

CIFAR-10 As a baseline, we evaluate ND methods on the CIFAR-10 dataset, which contains 10 common classes. This dataset has a large number of images per class (6000), but coarse image resolution (32x32 RGB images). In this

IN-125 Since prior work on ND has consisted mostly of testing on simple datasets (MNIST and CIFAR-10), in this work, we construct a reference dataset composed of 125 ImageNet classes (termed “IN-125”) on which we additionally evaluate the ND methods described above. Note that, by design, none of the 125 classes used here in IN-125 belongs to the original 1000 ImageNet competition classes that are used for training ConvNets like VGG16, so as to avoid issues that our ND problems may include outlier classes that in fact were used for pre-training weights of VGG-16 (or any other network used for discriminative embedding), which would invalidate the assumptions these classes have not been seen previously.

IN-125 increases complexity along two challenges when compared to ND experiments conducted on prior ND studies in that (a) the images’ resolutions are higher and (b) the number of exemplars per class is about one order of magnitude lower compared to datasets like CIFAR-10 or MNIST. We release the specification [11] of this dataset for future comparisons, which consists of the set of problems, each of which we detail by providing the list of classes used as inliers and outliers.

4. Experiments and performance characterization

We run experiments on the datasets detailed above. The results of applying our ND framework to two IN-125 problems is exemplified in Figures 2 and 3. We demonstrate the use of the performance evaluation framework we designed whereby we express performance, measured in terms of AUC, as a function of complexity for each of the approaches to ND described in this study. We also use the set of selected problems described in the previous section. For each problem we show the distribution of $AUC=f(\text{complexity})$.

We first performed complexity assessment using KLCA. Results suggested that it is not an effective empirical complexity measure in that KL divergence did not always correlate or predict the performance of ND algorithms. This is likely due to the issue of attempting to compute KL for distributions that don't have overlapping support everywhere, a problem which is exacerbated by the use of a limited number of points in high dimensional spaces.

We next evaluated performance as a function of BERCA and plot in Figs 4-7, for GAP512 features, whisker plots of the AUC as a function of the BERCA complexity, for each of the main ND scoring methods (1CSVM, LOF, IF and EE). The plots demonstrate the effectiveness of this approach in comparing methods. It can be observed that in general performance decreases as BERCA complexity increases for all methods, with graceful degradation, as should be expected. One exception to this is for LOF/ X_{GAP512} (Fig 7) which shows an average AUC that is less sensitive to increasing problem complexity (for the range of problems tested here). In aggregate, it is also notable that the resulting AUC performance is promising considering the problems' complexity.

Note that as a result of the large number of $n_c = 125$ classes in this case, we experimented on the MISO case of IN-125 only, with the following variation. We ran 125 experiments as expected, cycling the single outlier every time. However, for our inliers, we randomly chose 10 classes from the remaining $n_c - 1 = 124$ classes. By restricting the number of inliers in this way, we could perform a more accurate comparison to our CIFAR-10 results, without introducing unnecessary variables.

Next, we summarize AUC for all the methods in our framework on the different datasets in Table 1 showing average AUC over the problem considered. We also include results we obtain using best of breed generative methods including Ganomaly, DGAN and EGAN. We segregate results into discriminative and generative approaches because of their different nature and assumptions made (see more on this in the discussion). The best results are bold-faced. We can see clearly again the influence of complexity of the problem on the resulting method performance,

Figure 4. $AUC=f(c)$ (AUC as a function of problem complexity) for 125 problems in IN-125 (1CSVM applied to GAP features)

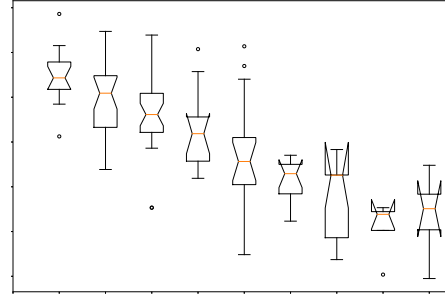


Figure 5. $AUC=f(c)$ (AUC as a function of problem complexity) for 125 problems in IN-125 (Elliptic Envelope applied to GAP features)

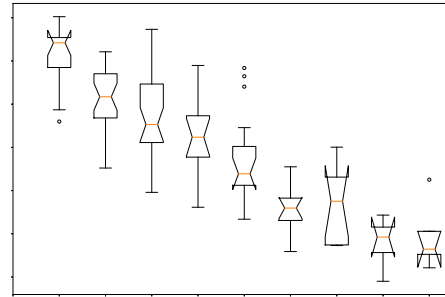
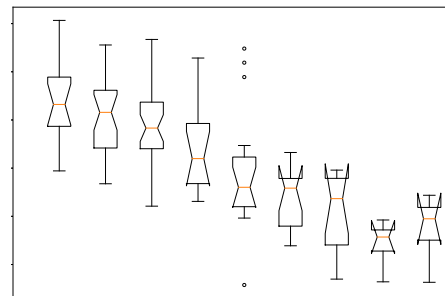
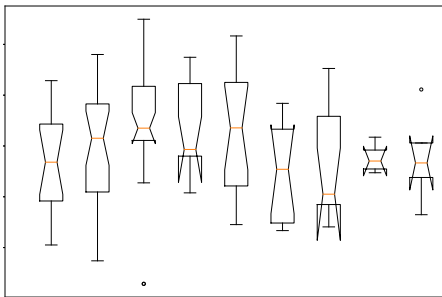


Figure 6. $AUC=f(c)$ (AUC as a function of problem complexity) for 125 problems in IN-125 (Isolation Forest applied to GAP features)



which was precisely characterized in earlier plots by using metric values of complexity via BERCA. In aggregate it can be seen that methods such as LOF/ X_{GAP512} and LOF/ X_{MULTI} perform best on CIFAR-10 (MISO), and EE/ X_{GAP512} and EE/ X_{MULTI} do so on the simpler set of problems in CIFAR-10 (SIMO), and that on 1CSVM/ X_{GAP512} and EE/ X_{GAP512} for the more complex

Figure 7. AUC=f(c) (AUC as a function of problem complexity) for 125 problems in IN-125 (Local Outlier Factor applied to GAP features)



IN-125.

For generative methods, the results show that the combination of 1CSVM and using X_{GAN} features improves upon the best in class methods previously reported for the class of generative ND methods [3].

5. Discussion

Analysis of results In general all deep embedding methods outperformed GAN-based methods. But these should not be taken as equal: this is because discriminative embeddings use pre-trained networks and as such exploit additional prior information in addition to the inlier training samples they used for training (which is the only information used by generative methods). This is why these methods should be considered separately. When considering only generative ND methods, one of the methods in our framework, combining X_{GAN} and 1CSVM, resulted in best overall performance in that class of methods and compared to prior work. In aggregate it is also encouraging to see that generative methods' performance, despite using less information, often comes close to that of discriminative methods.

How to best compare DL-based ND methods going forward? Recent ND studies using DL methods, even when they used the same datasets (e.g. MNIST), often did not adopt the same conventions for what classes were used as inliers or outliers, and different combinations and partitioning entailed different resulting problem complexity for the specific ND problems being addressed. There is therefore a pressing need for more comparable protocols when reporting ND performance methods as their performance significantly depend on the problem complexity. Because of this we see value in having future research studies use means of reporting performance of ND methods similar or inspired by what was adopted here: taking into account a quantitative complexity measure such as the one we proposed herein in BERCA, so as to allow apples-to-apples comparisons between studies. Showing trade spaces involv-

ing $AUC=f(\text{complexity})$ would also allow one to predict the operational performance of a method for new problems and datasets. Additionally, this approach has value for analyzing trends of results, as was shown in the whisker plots above.

Future work As an alternate approach to ND problem generation, it is possible to achieve different levels of complexity by selecting unknown instances with different levels of similarity to known classes based on class hierarchy or semantic relationship. Our proposed metric in essence takes that concept and introduces a more general quantitative measure that is tied more directly to classification performance. It would be valuable to investigate how our complexity measure is related to the complexity driven by subjective semantic similarity used in information retrieval. Our dataset can certainly facilitate such a study. Future/alternative ways to characterize complexity can be investigated: these could consist of using information theoretic measures such as KL-divergence with multivariate Gaussian assumptions or be inspired by metrics such as Bayesian theory of surprise [6]. Future work can also extend the use of ND for improved zero-shot learning performance [12, 30]

Recent work using GANs for embeddings and novelty detection has suggested possible benefits. However these studies were often conducted on restricted datasets, or datasets with small number of classes, large number of images per class, and/or low image resolution. This study shows that using GANs as representation confers benefits while using no prior information (other than inlier training data) even for more complex datasets. We believe that newer GAN formulations such as BigGAN [10] ProGAN [22] and StyleGAN [23] can play a role to further this work by allowing larger resolution images which in turn may entail latent spaces that would possibly facilitate better ND.

6. Conclusion

We present a framework for novelty detection based on deep embeddings, both discriminative and generative. We propose a new way to fairly characterize novelty detection performance using problem complexity which allows for apples-to-apples comparisons. One of the proposed generative methods in our framework demonstrates best of breed performance among recently proposed generative novelty detection methods.

Acknowledgements

We thank Jemima Albayda and Mauro Maggioni (JHU) for thoughtful inputs and discussions. The support of JHU APL internal research and development funding is gratefully acknowledged.

Table 1. Average AUC performance of various methods in our framework using generative or discriminative embeddings, and comparison with recent methods of record. In parenthesis: error margins for 95% confidence intervals.

Novelty detection methods via discriminative embeddings.			
Method (Novelty measure/Embedding vector)	CIFAR-10 (MISO)	CIFAR-10 (SIMO)	IN-125 (MISO)
1CSVM/ $X_{\text{GAP}512}$	0.5771 (0.1180)	0.6853 (0.0907)	0.6241 (0.1291)
1CSVM/ X_{MULTI}	0.5932 (0.0688)	0.7322 (0.0500)	0.5759 (0.1292)
EE/ $X_{\text{GAP}512}$	0.5408 (0.1130)	0.7196 (0.0931)	0.6280 (0.1458)
EE/ X_{MULTI}	0.5527 (0.0889)	0.7165 (0.0636)	0.5696 (0.1389)
IF/ $X_{\text{GAP}512}$	0.5378 (0.1168)	0.6706 (0.0817)	0.5437 (0.1238)
IF/ X_{MULTI}	0.5373 (0.0561)	0.6750 (0.0634)	0.5315 (0.1259)
LOF/ $X_{\text{GAP}512}$	0.6224 (0.0581)	0.6324 (0.0792)	0.5037 (0.1112)
LOF/ X_{MULTI}	0.6030 (0.0399)	0.6783 (0.0566)	0.5249 (0.1243)
Novelty detection methods via generative embeddings.			
Method (Novelty measure/Embedding vector)	CIFAR-10 (MISO)	CIFAR-10 (SIMO)	IN-125 (MISO)
1CSVM/ X_{GAN}	0.5627 (0.1458)	0.6279 (0.1266)	0.5792 (0.0719)
ND-DGAN [5, 36]	0.4898 (0.0305)	0.4495 (0.1125)	0.4789 (0.0300)
ND-EGAN [36, 41]	0.4655 (0.1250)	0.4183 (0.0950)	0.4822 (0.0741)
GANOMALY [3]	0.5321 (0.1292)	0.6228 (0.1092)	0.5708 (0.0775)

Appendix A: Additional Technical Details

We provide here some more details on the novelty detection algorithms: **LOF** The LOF of a point (here a feature vector X) is based on comparing the density of points around X against the density of X 's neighbours [9]. Defining first the k -distance $d^k(X)$ of X to that of its k -th nearest neighbour, and noting $L_k(X)$ the set of points (the so-called "MinPts") within $d^k(X)$, one defines the "reachability distance" of X from any origin point O and for a given k as: $R_k(X, O) = \max(d(X, O), d_k(O))$. To characterize density, one defines the local reachability density $\text{lrd}(X)$ by taking the inverse of the average reachability distance of all points $O \in L_k(X)$.

To compare densities, the LOF (X) is defined as the average of the $\text{lrd}(\cdot)$ of all points in $L_k(X)$ divided by $\text{lrd}(X)$. This ratio considers the average local densities of the neighbours of X compared to the local density of X .

Intuitively if this value is higher than one the point X is less dense (less reachable by its neighbours than the neighbours of X are by their own neighbours), and is therefore an outlier. Likewise, the opposite of the LOF can be used as a score to detect novelty (values of this increasing as a point becomes more an inlier).

1CSVM In 1CSVM, inliers points x are modeled as lying inside a hypersphere with center denoted a and radius R which is found by minimizing the error function:

$$F(R, a) = R^2 \text{ with } x_i - a^2 \leq R^2, \quad i \quad (1)$$

To allow for training datasets corrupted with outliers one introduces slack variables $\epsilon_i \geq 0$ to allow for some violations

$$x_i - a^2 \leq R^2 + \epsilon_i, \quad \epsilon_i \geq 0 \quad i \quad (2)$$

and modifies the minimization problem to include penalties on ϵ_i magnitude $F(R, a) = R^2 + C \sum \epsilon_i$, with C a weighting for slack variables. Using Lagrange multipliers $\alpha_i \geq 0$ and $\beta_i \geq 0$, this problem can be formulated as one of minimizing L with regard to R, a, x_i , and maximizing L with respect to α_i and β_i :

$$L(R, a, x_i, \alpha_i, \beta_i) = R^2 + C \sum \epsilon_i - \sum \alpha_i (x_i - a^2 - R^2) - \sum \beta_i \{R^2 + x_i - a^2\}$$

It can be shown [7, 19, 39] that this reduces to minimizing:

$$L = \sum_i (x_i \cdot x_i) - \sum_{i,j} \alpha_j (x_i \cdot x_j) \quad (3)$$

with constraints in Eq. (2). The linear dot product is commonly replaced with a nonlinear kernel $K(x, y)$ (e.g. RBF) to allow for nonlinear decision boundaries to emerge. Minimizing L produces a set of weights α_i for the corresponding samples x_i , and the center a and radius R of the hypersphere. By invoking the complementary slackness constraints training exemplars with nonzero weights emerge as *support vectors* of the data. To test out if a test exemplar y lies within the hypersphere, one can use as score the distance of a sample to the center of the hypersphere [7, 19, 39]:

$$d(y) = \frac{1}{R^2} [K(y, y) - 2 \sum_i \alpha_i K(y, x_i) + \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j)] \quad (4)$$

References

- [1] R. Abay, S. Gehly, S. Balage, M. Brown, and R. Boyce. Maneuver detection of space objects using generative adversarial networks. 2018. **1**
- [2] M. Ahmed, A. N. Mahmood, and M. R. Islam. A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems*, 55:278–288, 2016. **1**
- [3] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. *arXiv preprint arXiv:1805.06725*, 2018. **1, 2, 3, 7, 8**
- [4] L. Akoglu, H. Tong, and D. Koutra. Graph based anomaly detection and description: a survey. *Data mining and knowledge discovery*, 29(3):626–688, 2015. **1**
- [5] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017. **4, 8**
- [6] P. Baldi and L. Itti. Of bits and wows: A bayesian theory of surprise with applications to attention. *Neural Networks*, 23(5):649–666, 2010. **7**
- [7] A. Banerjee, P. Burlina, and C. Diehl. A support vector method for anomaly detection in hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 44(8):2282–2291, 2006. **1, 4, 8**
- [8] P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger, and C. Steger. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. *arXiv preprint arXiv:1807.02011*, 2018. **1**
- [9] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: identifying density-based local outliers. In *ACM sigmod record*, volume 29, pages 93–104. ACM, 2000. **4, 8**
- [10] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. **7**
- [11] P. Burlina, N. Joshi, and I.-J. Wang. Problem specification for in-125, url = <https://github.com/neil454/in-125>, urldate = 2019-03-19. **5**
- [12] P. M. Burlina, A. C. Schmidt, and I.-J. Wang. Zero shot deep learning from semantic attributes. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 871–876. IEEE, 2015. **7**
- [13] D. Carrera, G. Boracchi, A. Foi, and B. Wohlberg. Detecting anomalous structures by convolutional sparse models. In *Neural Networks (IJCNN), 2015 International Joint Conference on*, pages 1–8. IEEE, 2015. **1**
- [14] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009. **1**
- [15] G. Chen, M. Iwen, S. Chin, and M. Maggioni. A fast multi-scale framework for data in high-dimensions: Measure estimation, anomaly detection, and compressive measurements. In *Visual Communications and Image Processing (VCIP), 2012 IEEE*, pages 1–6. IEEE, 2012. **1**
- [16] L. Deecke, R. Vandermeulen, L. Ruff, S. Mandt, and M. Kloft. Anomaly detection with generative adversarial networks. 2018. **1, 2**
- [17] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. Wiley, New York, 1973. **4**
- [18] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie. High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning. *Pattern Recognition*, 58:121–134, 2016. **1**
- [19] D. E. Freund, N. Bressler, and P. Burlina. Automated detection of drusen in the macula. In *Biomedical Imaging: From Nano to Macro, 2009. ISBI'09. IEEE International Symposium on*, pages 61–64. IEEE, 2009. **8**
- [20] K. Gray, D. Smolyak, S. Badirli, and G. Mohler. Coupled igmm-gans for deep multimodal anomaly detection in human mobility data. *arXiv preprint arXiv:1809.02728*, 2018. **1**
- [21] N. Jain, L. Manikonda, A. O. Hernandez, S. Sengupta, and S. Kambhampati. Imagining an engineer: On gan-based data augmentation perpetuating biases. *arXiv preprint arXiv:1811.03751*, 2018. **1**
- [22] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. **7**
- [23] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. *arXiv preprint arXiv:1812.04948*, 2018. **7**
- [24] M. Kimura and T. Yanagihara. Semi-supervised anomaly detection using gans for visual inspection in noisy training data. *arXiv preprint arXiv:1807.01136*, 2018. **1**
- [25] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim. A survey of deep learning-based network anomaly detection. *Cluster Computing*, pages 1–13, 2017. **1**
- [26] Y. Lai, J. Hu, Y. Tsai, and W. Chiu. Industrial anomaly detection and one-class classification using generative adversarial networks. In *2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1444–1449. IEEE, 2018. **1**
- [27] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 413–422. IEEE, 2008. **4**
- [28] Y. Liu, Z. Li, C. Zhou, Y. Jiang, J. Sun, M. Wang, and X. He. Generative adversarial active learning for unsupervised outlier detection. *arXiv preprint arXiv:1809.10816*, 2018. **1**
- [29] J. Markowitz, A. C. Schmidt, P. M. Burlina, and I.-J. Wang. Hierarchical zero-shot classification with convolutional neural network features and semantic attribute learning. In *International Conference on Machine Vision Applications (MVA), 2017. IAPR*, 2017. **1**
- [30] J. Markowitz, A. C. Schmidt, P. M. Burlina, and I.-J. Wang. Hierarchical zero-shot classification with convolutional neural network features and semantic attribute learning. In *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*, pages 194–197. IEEE, 2017. **7**
- [31] S. Matteoli, M. Diani, and J. Theiler. An overview of background modeling for detection of targets and anomalies in hyperspectral remotely sensed imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(6):2317–2336, 2014. **1**
- [32] M. Naphade, M.-C. Chang, A. Sharma, D. C. Anastasiu, V. Jagarlamudi, P. Chakraborty, T. Huang, S. Wang, M.-Y. Liu, R. Chellappa, et al. The 2018 nvidia ai city challenge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 53–60, 2018. **1**

- [33] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko. A review of novelty detection. *Signal Processing*, 99:215–249, 2014. **1**
- [34] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. **2**
- [35] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 512–519. IEEE, 2014. **3**
- [36] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International Conference on Information Processing in Medical Imaging*, pages 146–157. Springer, 2017. **1, 2, 3, 8**
- [37] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. **3**
- [38] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943, 2013. **1**
- [39] D. M. Tax and R. P. Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004. **4, 8**
- [40] Q. Wang, S. R. Kulkarni, and S. Verdú. Divergence estimation for multidimensional densities via k-nearest-neighbor distances. *IEEE Transactions on Information Theory*, 55(5):2392–2405, 2009. **4**
- [41] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar. Efficient gan-based anomaly detection. *arXiv preprint arXiv:1802.06222*, 2018. **1, 2, 3, 8**