Generative Probabilistic Novelty Detection with Isometric Adversarial Autoencoders

Ranya Almohsen Matthew R. Keaton Donald A. Adjeroh Gianfranco Doretto
West Virginia University
Morgantown, WV 26506
{ralmohse, mrkeaton, daadjeroh, gidoretto}@mix.wvu.edu

Abstract

Learning the manifold of a complex distribution is a fundamental challenge for novelty or anomaly detection. We introduce a revised learning and inference procedure that takes into account a key underlying assumption made by the framework of generative probabilistic novelty detection. The traditional framework implies the ability to not only learn the manifold of the generative distribution of inliers but also to compute non-linear orthogonal projections onto this manifold from the ambient space. We augment the original training with priors that endow the model with this property, and prove that inference becomes easier and computationally more efficient. We show experimentally that the new framework leads to improved and more stable results.

1. Introduction

The task of recognizing data samples to be inliers or outliers is often referred to as novelty or anomaly detection, and is an important process in a number of fields related to research, medicine, and industry [43]. In many cases, novelty detection can be a crucial step (e.g., in open-set recognition [45]), and it is thus valuable finding and optimizing approaches to this problem. Often, the task is framed as one of learning an inlier distribution and determining the likelihood of a new sample belonging to it. Thus, producing a generalizing distribution model from a finite set of training samples, and inferring likelihoods, become the central challenges to many approaches, especially in computer vision, where data samples are high-dimensional.

In this work, we introduce a new method that stems from addressing a few existing weaknesses of our previous approach known as Generative Probabilistic Novelty Detection (GPND) [37]. We revise the derivation of the novelty/anomaly test, where we highlight and make further use of the central hypothesis of computing non-linear orthogonal projections from the ambient space, where outliers come from, onto the manifold where inliers live. Doing so leads to important computational improvements because we prove that the need for computing costly Jacobians during inference is completely removed. On the training side, we show that this entails learning a parameterized inlier manifold that is an isometry, while we also need to learn a mapping that projects from the ambient space onto the manifold and then inverts the isometry. We show that we can implement our model with adversarial autoencoders where we add specialized priors for learning such isometry and pseudo-inverse maps. As a byproduct, this learning approach lends to smoother manifolds, thus more likely to generalize well, which is vital to the process of determining the inlier distribution.

In the rest of the paper we review the related work in Section 2. We revise the formulation of GPND in Section 3. In Section 4 we complete the new formulation with learning the isometry and the pseudo-inverse maps, and describe the architecture, priors and losses to do so. In Section 5 we analyze in detail the performance of our new approach, which we name *Generative Probabilistic Novelty Detection with Isometry (GPNDI)*.

2. Related Work

In this section, we summarize the literature for novelty detection as well as other related topics including out-of-distribution detection. Novelty detection methods can generally be split into three overarching groups: probabilistic, density estimation, and reconstruction-based methods.

Traditional probabilistic methods [1, 8, 18, 59] estimate the probability density function of normal data points by inferring the model parameters. New data points with the smallest likelihood are identified as outliers. A popular approach uses the Gaussian Mixture Model (GMM), which fits a selected number of Gaussian distributions to a dataset using the Expectation-Maximization (EM) algorithm [22]. GMM has been used in applications including the identification of suspicious and possibly cancerous masses in mammo-

grams [51]. Additionally, kernel-based probabilistic methods learn the null space of training data and rely on distance measures to perform density estimation implicitly [2, 14, 27, 62]. Our approach relates to these approaches because it derives a novelty test following the same likelihood principle.

Density estimation methods include a recent category of approaches such as DifferNet [42], which adopts the normalizing flow [40] as a density estimation of the image features extracted by convolutional neural networks. The anomaly score is then computed based on the likelihoods of multiple transformations per image. Other normalizing flow-based methods include [11, 41, 61]. Each of them contains two main components: the feature extraction module and the distribution estimation module. An advantage of these models over other methods is that one can calculate the likelihood of a point directly without any approximation while also being able to sample from it reasonably efficiently. However, evaluating each layer's Jacobian and its determinant can be very expensive and slow at test time, especially with high-dimensional data [7, 15]. Another drawback of these methods is that they do not perform any dimensionality reduction [33], which makes them less useful with high dimensional data like images. Our approach relates to these approaches but it overcomes both of these drawbacks by eliminating the need for computing Jacobians during inference, and by performing dimensionality reduction since the dimension of the inlier manifold is much smaller than the dimension of the ambient space.

Reconstruction-based methods tend to utilize generative models like auto-encoders or generative adversarial networks [9] to encode and reconstruct the normal data. [12, 56] used deep learning-based autoencoders to learn the model of normal behaviors and employed a reconstruction loss to detect outliers. [53] used a GAN-based method, where the generator is used to recover a latent representation with gradient descent, by optimizing upon the reconstruction error, which was then used as a novelty score. [39] trained GANs using optical flow images to learn a representation of scenes in videos. [54] minimized the reconstruction error of an autoencoder to remove outliers from noisy data, and by utilizing the gradient magnitude of the autoencoder they make the reconstruction error more discriminative for positive samples. In [44], a framework was proposed for one-class classification and novelty detection. It consists of two main modules learned in an adversarial fashion. The first is a decoder-encoder convolutional neural network trained to reconstruct inliers accurately, while the second is a one-class classifier made with another network that produces the novelty score.

Computing reconstruction error in image space is not ideal, and in fact, the L_2 norm works poorly with images. [19] used as a novelty score not only the reconstruction error in the image space, but also in hidden spaces. They pass the reconstructed image to the encoder and observe activations of all the intermediate layers in the encoder and compare those to activations induced by the original image. [46] extended this approach by adding an adversarial loss that matches the distribution of hidden activations for real and reconstructed inliers. In [36] DCAE exclusively reconstructs the in-class data by learning their latent representations to be compact and collapse-free. DCAE utilizes its own internal module that captures class semantics of the in-class data for both effective training and inference. Our approach relates to these because it learns a generative model of the data, but during inference we use it to directly compute the likelihood of datapoints, rather than a novelty or anomaly score.

Out-of-Distribution methods usually improve robustness of existing classification or detection systems in order to detect erroneous samples (i.e., from other problem domains or datasets) that otherwise would be classified incorrectly. A recent line of work has focused on detecting out-of-distribution samples by analyzing the output entropy of a prediction made by a pre-trained deep neural network [6, 13, 17, 25, 28, 49, 50]. This is done by either simply thresholding the maximum softmax score [13] or by first applying perturbations to the input, scaled proportionally to the gradients with respect to the input and then combining the softmax score with temperature scaling, as it is done in Out-of-distribution Image Detection in Neural Networks (ODIN) [25]. While these approaches require labels for the in-distribution data to train the classifier network, our method does not use label information.

3. Generative Probabilistic Novelty Detection

Here we revise the derivation of the formulation of the novelty/anomaly test initially introduced in [37] to emphasize certain properties that were previously untapped, and to keep the paper self-contained. Specifically, we assume that training data points $x_1, \ldots, x_N$, where $x_i \in \mathbb{R}^m$, are sampled, possibly with noise ξ_i, from the model

$$x_i = f(z_i) + \xi_i \qquad i = 1, \cdots, N \,, \tag{1}$$

where z_i is defined in a *latent* space $\Omega \subset \mathbb{R}^n$. The mapping $f : \Omega \to \mathbb{R}^m$ defines $\mathcal{M} \equiv f(\Omega)$, which is a parameterized manifold of dimension n, with $n < m$. We also assume that the Jacobi matrix of f is full rank at every point of the manifold.

Given a new data point $\bar{x} \in \mathbb{R}^m$, we design a novelty test to assert whether $\bar{x}$ was sampled from model (1). We begin by computing the non-linear orthogonal projection of $\bar{x}$ onto $\mathcal{M}$, which we indicate as $\bar{x}^{\|} \in \mathcal{M}$, and that in latent space is given by $\bar{z}$, where $\bar{x}^{\|} = f(\bar{z})$, and

$$\bar{z} = \arg\min_{z} \|\bar{x} - f(z)\| \,, \tag{2}$$

in which $\|\cdot\|$ is the L_2 norm. Assuming f to be smooth enough, we perform a linearization around $\bar{z}$, based on its

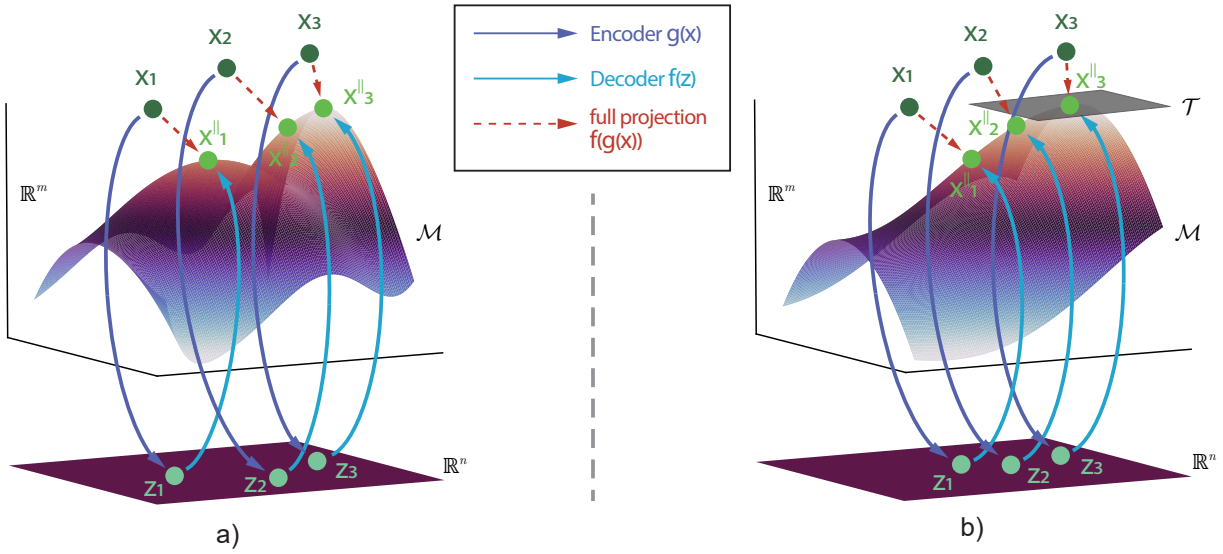

Figure 1. **Isometric manifold schematic representation.** Improving on the efforts of GPND, (a), isometric autoencoders, (b), enforce an angle and distance-preserving mapping from $\mathbb{R}^m$ to the low-dimensional manifold $\mathcal{M} \subset \mathbb{R}^m$ and then onto the latent space $\mathbb{R}^n$. Additionally, the full mapping $f(g(x))$ is encouraged to be orthogonal to $\mathcal{M}$, and therefore to the tangent space $\mathcal{T}$. Constraining the learned manifold in this manner generally lends to a smoother mapping and more appropriate generalization of the training data.

first-order Taylor expansion

$$f(z) = f(\bar{z}) + J_f(\bar{z})(z - \bar{z}) + O(\|z - \bar{z}\|^2) \,, \quad (3)$$

where $J_f(\bar{z})$ is the Jacobi matrix computed at $\bar{z}$. We note that $\mathcal{T} = \text{span}(J_f(\bar{z}))$ represents the tangent space of $\mathcal{M}$ at $\bar{x}^{\|}$ that is spanned by the n independent column vectors of $J_f(\bar{z})$, see Figure 1(b). Also, we have $\mathcal{T} = \text{span}(U^{\|})$, where $J_f(\bar{z}) = U^{\|} S V^\top$ is the singular value decomposition (SVD) of the Jacobi matrix. The matrix $U^{\|}$ has rank n, and if we define $U^\perp$ such that $U = [U^{\|} U^\perp]$ is a unitary matrix, we can represent the data point $\bar{x}$ with respect to the coordinates that are parallel to the tangent space $\mathcal{T}$, and to its orthogonal complement $\mathcal{T}^\perp$. This is done by computing

$$\bar{w} = U^\top \bar{x} = \begin{bmatrix} {U^{\|}}^\top \bar{x} \\ {U^\perp}^\top \bar{x} \end{bmatrix} = \begin{bmatrix} \bar{w}^{\|} \\ \bar{w}^\perp \end{bmatrix} \,, \quad (4)$$

where the rotated coordinates $\bar{w}$ are decomposed into $\bar{w}^{\|}$, which are parallel to $\mathcal{T}$, and $\bar{w}^\perp$ which are orthogonal to $\mathcal{T}$.

We now indicate with $p_X(x)$ the probability density function describing the random variable X, from which training data points have been drawn. Also, $p_W(w)$ is the probability density function of the random variable W representing X after the change of coordinates (4). The two distributions are identical modulo the coordinate change. However, we make the assumption that the coordinates $W^{\|}$, which are parallel to $\mathcal{T}$, and the coordinates $W^\perp$, which are orthogonal to $\mathcal{T}$, are statistically independent. This means that in a neighborhood of $\bar{x}^{\|}$, the following holds

$$p_X(x) = p_W(w) = p_W(w^{\|}, w^\perp) = p_{W^{\|}}(w^{\|}) p_{W^\perp}(w^\perp) \,. \quad (5)$$

This is motivated by the fact that in (1) the noise ξ is assumed to predominantly deviate the point x away from the manifold $\mathcal{M}$ in a direction orthogonal to $\mathcal{T}$. This means that $W^\perp$ is primarily responsible for the noise effects, and since noise and drawing from the manifold are statistically independent, so are $W^{\|}$ and $W^\perp$.

From (5), given a new data point $\bar{x}$, we propose to perform novelty detection by executing the following test

$$p_X(\bar{x}) = p_{W^{\|}}(\bar{w}^{\|}) p_{W^\perp}(\bar{w}^\perp) = \begin{cases} \geq \gamma & \Longrightarrow \text{ Inlier} \\ < \gamma & \Longrightarrow \text{ Outlier} \end{cases} \quad (6)$$

where γ is a suitable threshold.

3.1. Data distribution learning and inference

The novelty detector (6) requires the computation of $p_{W^{\|}}(w^{\|})$ and $p_{W^\perp}(w^\perp)$. Here we provide a revised version from [37], of the description of how these distributions can be learned from data, and used for inference, with some differences that exploit the fact that it is possible to compute precise geometric projections from the ambient space $\mathbb{R}^n$ onto $\mathcal{M}$, via (2). We note that $w^{\|}$ can be written as $w^{\|} = {U^{\|}}^\top x = {U^{\|}}^\top (x - x^{\|}) + {U^{\|}}^\top x^{\|} = {U^{\|}}^\top x^{\|}$, where ${U^{\|}}^\top (x - x^{\|}) = 0$, because $x - x^{\|}$ is orthogonal to the tangent space $\mathcal{T}$. Therefore, $w^{\|}$ and z are related as

$w^{\|} = {U^{\|}}^{\top} f(z)$. Let us now indicate with Z the random variable representing the latent space, and with $p_Z(z)$ its probability distribution. By using the linearization (3), and the fact that V is a unitary matrix, it is easy to realize that $p_Z(z)$ and the distribution $p_{W^{\|}}(w^{\|})$, around the neighborhood of $f(z)$, are related as follows

$$p_{W^{\|}}(w^{\|}) = |\det S^{-1}|\, p_Z(z) \ . \tag{7}$$

Note that $p_Z(z)$ is independent from the linearization and it can be learned offline. Specifically, from the training data $\{x_i\}$, we compute their orthogonal projections in the latent space $\{z_i\}$ according to (2), and we fit to them a generalized Gaussian distribution to represent $p_Z(z)$ with a parametric model.

In order to compute $p_{W^{\perp}}(w^{\perp})$, we approximate it with its average over the hypersphere $\mathcal{S}^{m-n-1}$ of radius $\|w^{\perp}\|$, giving rise to

$$p_{W^{\perp}}(w^{\perp}) \approx \frac{\Gamma\left(\frac{m-n}{2}\right)}{2\pi^{\frac{m-n}{2}} \|w^{\perp}\|^{m-n-1}} p_{\|W^{\perp}\|}(\|w^{\perp}\|) \ , \tag{8}$$

where $\Gamma(\cdot)$ represents the gamma function. This is motivated by the fact that noise of a given intensity will be equally present in every direction.

Computing (8) requires $p_{\|W^{\perp}\|}(\|w^{\perp}\|)$, which is the distribution of the norms of $w^{\perp}$, and in principle, it could easily be learned offline by histogramming the norms of $w^{\perp} = {U^{\perp}}^{\top} x$, computed for each of the training data points $\{x_i\}$. On the other hand, the same distribution can be learned even more easily, without the need for computing the Jacobi matrix at each point, by observing the following. Since $x - x^{\|}$ is orthogonal to $\mathcal{T}$, it means that $x^{\|}$ is orthogonal to $\mathcal{T}^{\perp}$, i.e., ${U^{\perp}}^{\top} x^{\|} = 0$. Therefore, we have that $w^{\perp} = {U^{\perp}}^{\top} x = {U^{\perp}}^{\top} x - {U^{\perp}}^{\top} x^{\|} = {U^{\perp}}^{\top}(x - x^{\|})$. Moreover, by taking the squared norms, we can also write that

$$\begin{aligned} \|w^{\perp}\|^2 &= \|{U^{\perp}}^{\top}(x - x^{\|})\|^2 + \|{U^{\|}}^{\top}(x - x^{\|})\|^2 \\ &= \|U^{\top}(x - x^{\|})\|^2 = \|x - x^{\|}\|^2 \ , \end{aligned} \tag{9}$$

where the last equality follows from U being unitary. If we define $x^{\perp} \doteq x - x^{\|}$, this means that in (8), we can replace $\|w^{\perp}\|$ with $\|x^{\perp}\|$, and $p_{\|W^{\perp}\|}(\|w^{\perp}\|)$ with $p_{\|X^{\perp}\|}(\|x^{\perp}\|)$. $x^{\perp}$ does not require the Jacobi matrix to be computed, making the learning and inference more efficient. Specifically, $p_{\|X^{\perp}\|}(\|x^{\perp}\|)$ is learned from the training data $\{x_i\}$, by computing their orthogonal projections according to (2), and histogramming the L_2 norms between data and projectons.

4. Manifold learning

A major task in our approach is to learn the manifold $\mathcal{M}$. Here we derive the requirements, the network architectures, and the set of losses needed to do the learning.

4.1. Model driven requirements

The manifold $\mathcal{M}$ is parameterized by the mapping f. Out of all the possible choices we propose to learn an isometric map. Imposing f to be an *isometry* is beneficial for multiple reasons. First, we do not loose representational power as long as $m \geq n + 1$ [34]. Second, it is easy to realize that if two isometries can represent $\mathcal{M}$, then they must be related by a rigid transformation [10]. This reduces the search space for the mapping f, and from a learning perspective, imposing this restriction will act positively, as a regularizer by reducing the hypotheses space.

On the other hand, the most important reason for f to be an isometry is that the Jacobian $J_f(z)$ will have orthonormal columns. This means that

$$J_f(z)^{\top} J_f(z) = I \ , \tag{10}$$

where I is the identity matrix. Therefore, it follows that $|\det S| = 1$, where S is the matrix with the singular values of the Jacobian, i.e., that (7) reduces to

$$p_{W^{\|}}(w^{\|}) = p_Z(z) \ , \tag{11}$$

since $|\det S^{-1}| = 1$. This result has very important computational implications, because it means that to compute the detection test (6) it will not be necessary to compute the Jacobian $J_f(z)$, which is by far the most time consuming step in the original GPND [37], not to mention that it introduces significant noise in evaluating the sample probability.

Finally, we note that this updated framework has some parallels with very elegant recent work on novelty/anomaly detection [32], which is based on computing probabilities via normalizing flows [20]. While backed by a clear theoretical framework, these approaches require computing the inverse Jacobian at every network layer, leading to major computational drawbacks. The issue stems in part from the fact that in computing the latent representation they do not perform dimensionality reduction. On the other hand, this updated GPND formulation not only does it learn a reduced representation, it also eliminates the need to compute Jacobians.

In order to compute the test (6), we need to have the representation z, as required by (11). According to (2), this can be done by first applying to $\bar{x}$ an orthogonal projection $P_{\mathcal{M}}$ from the ambient space onto $\mathcal{M}$, and then map the projection to the representation space via f^{-1}. This means that besides the manifold representation f, we also need to learn a function g, defined as

$$g(x) \doteq f^{-1} \circ P_{\mathcal{M}}(x) \ . \tag{12}$$

Figure 1 depicts this sequence of transformations. It can be shown, as described in [10], that if f is an isometry, then g

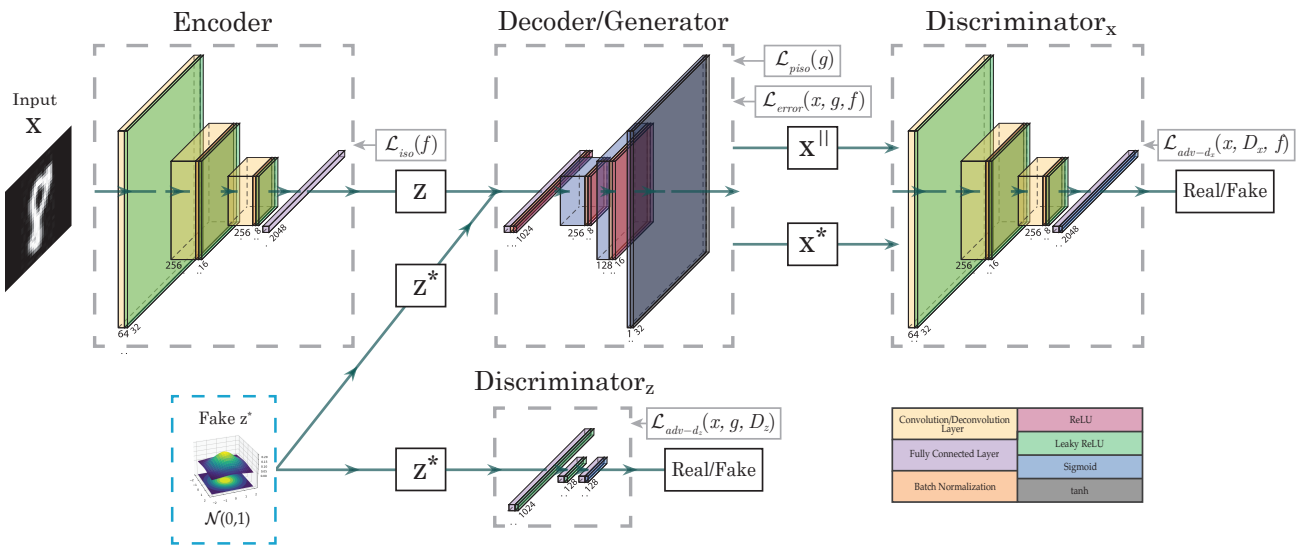

Figure 2. **Architecture overview.** Architecture of the network for manifold learning. It is based on training an Adversarial Autoenconder (AAE) [29]. Similarly to [3, 44] it has an additional adversarial component to improve generative capabilities of decoded images and a better manifold learning. The architecture layers of the AAE and of the discriminator D_x are specified on the right. All fake samples are generated from an n-dimensional normal distribution $\mathcal{N}(0, 1)$. x^* represents a projection of z^* onto the learned manifold $\mathcal{M}$.

is such that

$$J_g(f(z))J_g(f(z))^\top = I \,, \tag{13}$$

$$J_g(f(z)) = J_f(z)^\top \,. \tag{14}$$

Therefore, in order to compute (6) we need to learn two functions f and g which behave according to (10), (13), and (14). We stress the fact that satisfying all these requirements is fundamental, because the revised GPND framework is based on being able to compute (2), which also leads to the simplification (9), with the complete elimination of the need to compute Jacobians during testing.

4.2. Training losses

We plan to learn f and g with an autoencoder architecture, since for data points on the manifold the reciprocity must be satisfied, i.e., $x = f(g(x))$, but we require also (10), (13), and (14) to be satisfied as well. To that end, we build on the approach in [10, 16], and incorporate the following priors to the original GPND framework [37]. The first prior is the isometry loss $\mathcal{L}_{iso}(f)$, which encourages (10), and is defined as

$$\mathcal{L}_{iso}(f) = E\left[(\|J_f(z)u\| - 1)^2\right] \tag{15}$$

where $E[\cdot]$ denotes expectation, and u is uniformly sampled from the unit-sphere of dimension $n-1$, i.e., $\mathcal{S}^{n-1} = \{u \in \mathbb{R}^n \mid \| u \| = 1\}$.

The second prior is the pseudo-inverse loss $\mathcal{L}_{piso}(g)$, which encourages (13), and is defined as

$$\mathcal{L}_{piso}(g) = E\left[(\|u^\top J_g(x)\| - 1)^2\right] \tag{16}$$

where, again, u is sampled from $\mathcal{S}^{n-1}$. We combine these priors in this notation

$$\mathcal{L}_{iso\_AE}(f, g) = \mathcal{L}_{iso}(f) + \mathcal{L}_{piso}(g) \tag{17}$$

For the implementation of the prior above we follow the same strategy described in [10].

The backbone architecture mimics the adversarial autoencoder design in [37]. One adversarial component imposes a prior distribution on the latent space, the output of the encoder, that is matched with a normal distribution $\mathcal{N}(0, 1)$. The second adversarial component matches the output distribution of the decoder with the distribution of real data, representing the manifold $\mathcal{M}$. Finally, a cross-entropy loss is used to impose the reciprocity of the autoencoder, which is also the loss responsible to for encouraging (14), as discussed in [10].

The network architecture is shown in Figure 2. The adversarial losses are summarized as follows for the two adversarial components:

$$\begin{aligned}\mathcal{L}_{adv-d_z}(x, g, D_z) = E[\log(D_z(\mathcal{N}(0,1)))] \\ + E[\log(1 - D_z(g(x)))] \,,\end{aligned} \tag{18}$$

$$\begin{aligned}\mathcal{L}_{adv-d_x}(x, D_x, f) = E[\log(D_x(x))] \\ + E[\log(1 - D_x(f(\mathcal{N}(0,1))))] \,,\end{aligned} \tag{19}$$

Instead, $\mathcal{L}_{error}$ is used to minimize the reconstruction error for the input x that belongs to the known data distribution.

$$\mathcal{L}_{error}(x, g, f) = -E_z[\log(p(f(g(x))|x))] \,, \tag{20}$$

For simplicity, we combine all the losses without discriminators in $\mathcal{L}_{auto\_error}$, so that

$$\mathcal{L}_{auto\_error}(x, g, f) = \lambda_{iso}\mathcal{L}_{iso\_AE}(f, g) + \mathcal{L}_{error} \quad (21)$$

Where λ_{iso} is a hyperparameter for balancing the losses. Therefore, our objective function is going to be

$$\begin{aligned}\mathcal{L}(x, g, D_z, D_x, f) =& \mathcal{L}_{adv-d_z}(x, g, D_z) + \\ & \mathcal{L}_{adv-d_x}(x, D_x, f) + \\ & \lambda\mathcal{L}_{auto\_error}(x, g, f)\ , \end{aligned} \quad (22)$$

where λ is a hyper parameter that adjusts the trade off between the losses with and without discriminators. The autoencoder network is obtained by minimizing equation (22), giving:

$$\hat{g}, \hat{f} = \arg\min_{g,f} \max_{D_x, D_z} \mathcal{L}(x, g, D_z, D_x, f)\ . \quad (23)$$

We trained the proposed model by using stochastic gradient descent and doing alternative updates of each component as follows

- Maximize $\mathcal{L}_{adv-d_x}$ by updating weights of D_x;
- Minimize $\mathcal{L}_{adv-d_x}$ by updating weights of f;
- Maximize $\mathcal{L}_{adv-d_z}$ by updating weights of D_z;
- Minimize $\mathcal{L}_{auto\_error}$ and $\mathcal{L}_{adv-d_z}$ by updating weights of g and f.

5. Experiments

In this section, we present the set of experiments that have been conducted to demonstrate the effectiveness of our method. The performance results are analyzed in detail and are compared with state-of-the-art techniques where each of the results were taken from the original papers. In all cases, experiments are carried out identically to GPND [37].

For each experiment, datasets are randomly split into training, validation, and testing sets. In this setting, we do not reuse the same inliers for training and testing to make our evaluation more realistic. We compare our results to a few other approaches, namely [4, 44, 54] that do not follow this protocol and instead use the same inliers for training and testing.

Performance of our approach is evaluated using the F_1 measure, area under the ROC (AUROC), false positive rate (FPR) at 95% true positive rate (TPR), Detection Error at 95% TPR, and area under the precision-recall curve, calculated in terms of inliers (AUPR-In) and outliers (AUPR-Out).

Table 1. F_1 scores on MNIST [23]. Inliers are taken to be images of one category, and outliers are randomly chosen from other categories. All results are averages from a 5-fold cross validation.

% of outliers	$\mathcal{D}(\mathcal{R}(X))$ [44]	$\mathcal{D}(X)$ [44]	LOF [4]	DRAE [54]	GPND [37]	GPNDI (Ours)
10	0.97	0.93	0.92	0.95	0.983	**0.984**
20	0.92	0.90	0.83	0.91	0.971	**0.976**
30	0.92	0.87	0.72	0.88	0.961	**0.968**
40	0.91	0.84	0.65	0.82	0.950	**0.960**
50	0.88	0.82	0.55	0.73	0.939	**0.953**

Table 2. Results on Fashion-MNIST [55]. F_1 scores where inliers are taken to be images of one category, and outliers are randomly chosen from other categories.

% of outliers	10	20	30	40	50
GPND [37]	0.968	0.945	0.917	0.891	0.864
GPNDI (Ours)	**0.972**	**0.974**	**0.930**	**0.904**	**0.873**

5.1. Datasets

We evaluate our method on MNIST, Fashion-MNIST, Coil-100, CIFAR-10, and CIFAR-100.

MNIST [23] is composed of 70,000 28×28 handwritten digits.

Fashion-MNIST [55] contains 70,000 28×28 grayscale images of fashion items. Like MNIST, there are 10 categories each possessing 7,000 total samples.

Coil-100 [35] is comprised of 7,200 images. For each of 100 objects, pictures were taken 5 degrees apart from one another, resulting in 72 images for each object.

CIFAR-10 and CIFAR-100 [21] each possess 60,000 32×32 images with 10 and 100 classes, respectively. Both datasets contain a variety of balanced classes ranging from vehicles to animals, although no classes are shared between them. Like GPND [37] and ODIN [25], we count inliers as samples from either dataset, while images from two different cropped and resized versions of both TinyImageNet [5] and LSUN [48] are used individually as outliers. During validation, we use samples from iSUN [58] as outliers. We reuse the currently available datasets provided by ODIN's GitHub project page.

5.2. Implementation details and complexity

Since our implementation was done based on the source code of GPND, we follow most details with some differences related to hyperparameter values. We learn the isometric mapping by training g, and f while imposing the described specifications. During testing we do not need to compute any derivatives, such as the Jacobian matrix, which makes our approach significantly more efficient. Training is done with ADAM optimizer, we train the model for 100 epochs, λ_{iso} was set to 0.01, using an NVIDIA TITAN RTX.

Table 3. Results on Coil-100. Inliers are taken to be images of one, four, or seven randomly chosen categories, and outliers are randomly chosen from other categories (at most one from each category).

	OutRank [30, 31]	CoP [38]	REAPER [24]	OutlierPursuit [57]	LRR [26]	DPCP [52]	ℓ_1 thresholding [47]	R-graph [60]	GPND [37]	**GPNDI (Ours)**
Inliers: **one** category of images , Outliers: 50%										
AUC	0.836	0.843	0.900	0.908	0.847	0.900	0.991	**0.997**	0.968	0.984
F1	0.862	0.866	0.892	0.902	0.872	0.882	0.978	**0.990**	0.979	0.894
Inliers: **four** category of images , Outliers: 25%										
AUC	0.613	0.628	0.877	0.837	0.687	0.859	0.992	**0.996**	0.945	0.960
F1	0.491	0.500	0.703	0.686	0.541	0.684	0.941	**0.970**	0.960	0.953
Inliers: **seven** category of images , Outliers: 15%										
AUC	0.570	0.580	0.824	0.822	0.628	0.804	0.991	**0.996**	0.919	0.950
F1	0.342	0.346	0.541	0.528	0.366	0.511	0.897	0.955	0.941	**0.964**

Table 4. CIFAR-10 (CIFAR-100) comparison with ODIN [25] and GPND [37]. ↑ indicates larger value is better, and ↓ indicates lower value is better.

	Outlier dataset	**FPR(95%TPR)↓**	**Detection↓**	**AUROC↑**	**AUPR in↑**	**AUPR out↑**
		ODIN-WRN-28-10 / ODIN-Dense-BC / GPND / GPNDI (Ours)				
CIFAR-10	TinyImageNet (crop)	23.4/**4.3**/29.1/26.6	14.2/**4.7**/15.7/14.1	94.2/**99.1**/90.1/93.4	92.8/**99.1**/84.1/85.2	94.7/99.1/**99.5**/95.1
	TinyImageNet (resize)	25.5/**7.5**/11.8/22.7	15.2/**6.3**/8.3/24.6	92.1/**98.5**/96.5/97.1	89.0/**98.6**/95.0/88.1	93.6/98.5/**99.8**/89.2
	LSUN (crop)	21.8/**8.7**/89.1/61.1	13.4/**6.9**/47.0/22.6	95.9/**98.2**/35.8/96.0	95.8/**98.5**/39.1/81.6	95.5/**97.8**/83.7/85.3
	LSUN (resize)	17.6/**3.8**/4.9/5.6	11.3/**4.4**/4.9/5.1	95.4/**99.2**/98.7/98.9	93.8/**99.3**/98.4/97.2	96.1/99.2/**99.7**/98.1
CIFAR-100	TinyImageNet (crop)	43.9/**17.3**/33.2/32.1	24.4/**11.2**/17.2/23.0	90.8/**97.1**/89.1/90.6	91.4/**97.4**/83.8/88.1	90.0/96.8/**98.7/98.8**
	TinyImageNet (resize)	55.9/44.3/**15.0** /26.4	30.4/24.6/**9.5**/23.9	84.0/90.7/ 95.9/**96.1**	82.8/91.4/**94.6**/89.2	84.4/90.1/**99.4**/86.4
	LSUN (crop)	39.6/**17.6**/91.3/60.7	22.3/**11.3**/48.1/24.3	92.0/**96.8**/35.0/92.3	92.4/**97.1** /38.8/81.6	91.6/**96.5**/79.4/82.1
	LSUN (resize)	56.5/44.0/**6.8**/69.4	30.8/24.5/**5.8**/26.3	86.0/91.5/**98.3**/88.0	86.2/92.4/**98.0**/79.2	84.9/90.6/**99.6**/78.1

5.3. Results

MNIST dataset. For this set of tests, we create 5 random data splits with a balanced number of samples per class. We then evaluate on one split, using three of the remaining four for training and the final split for validation. The value of γ that produces the highest F_1 score on the validation set is then used during testing. Experiments are run using each digit as an inlier while the remaining digit samples are selected in order to generate outlier percentages between 10% to 50%. Results are shown in Table 1, and Figure 3. They suggest that GPNDI performs significantly better than GPND.

Fashion-MNIST dataset. We repeat the same protocol that we have used for MNIST, but on Fashion-MNIST. Results are provided in Table 2. We compare our results with GPND [37]. All results are averages from a 5-fold cross-validation. Our proposed method exceeds GPND in all cases.

Coil-100 dataset. Similar to the datasets above, we create five even splits for cross-validation, but instead use four for training and one for testing. The optimal γ is thus found on the training set. For each experiment, 1, 4, or 7 classes are randomly chosen to be inliers while the remaining classes are considered outliers and are included at some pre-selected percentage.

Results on Coil-100 are shown in Table 3. This table confirms that our method achieves a higher AUROC than GPND [37] in all cases. Interestingly, our method yields the highest F_1 score when the number of inlier categories is increased to seven, which shows that our method robustly learns the inlier representation and does not deteriorate as the number of inlier categories increase. We do not outperform R-graph [60] as they use a pre-trained VGG network, and we use an autoencoder with a very small architecture that we train from scratch on a very limited number of samples, which is on average only 70 per category.

CIFAR-10 (CIFAR-100) dataset. The available datasets at the time of publication were both versions of TinyImageNet and LSUN, as well as iSUN. For each experiment, samples for iSUN were used as outliers during validation, while for testing we use each of the remaining datasets as outliers. We report these results in Table 4, the performance of our method is compared with ODIN [25] and GPND [37]. In many cases, such as with CIFAR-10, LSUN (crop), and CIFAR-100, LSUN (crop), GPNDI performs better than GPND. We do not outperform ODIN but in some cases we do, such as with AUPR out of CIFAR-100,TinyImageNet (crop), and AUROC of CIFAR-100, and TinyImageNet (resize). This is still noticeable because ODIN requires label information provided with the training samples and uses a deep network with more than 100 layers (Dense-BC and WRN), whereas in our settings GPNDI dose not have any training label information and uses a very small auto-encoder network.

Table 5. MNIST comparison with baselines. All values are percentages. ↑ indicates larger value is better, and ↓ indicates lower value is better.

	10%	20%	30%	40%	50%	10%	20%	30%	40%	50%	10%	20%	30%	40%	50%
	F1↑					AUROC↑					FPR(95%TPR)↓				
Ours	**98.4**	**97.6**	**96.8**	**96.0**	**95.3**	**98.8**	**98.8**	**98.8**	**98.8**	**98.8**	**0.060**	**0.056**	**0.057**	**0.060**	**0.057**
GPND	98.2	97.1	96.1	95.0	93.9	98.1	98.0	98.0	98.0	98.0	8.1	9.1	8.7	8.8	8.9
AE	84.8	79.6	79.5	77.6	75.6	93.4	93.8	93.4	92.9	92.8	24.3	24.6	24.7	23.9	23.7
P-VAE	97.6	95.8	94.2	92.4	90.5	95.2	95.7	95.6	95.8	95.9	18.8	18.0	17.4	17.3	17.0
P-AAE	97.3	95.5	94.0	92.0	90.2	95.2	95.6	95.3	95.2	95.3	20.7	19.3	19.0	18.9	18.6
	Detection error↓					AUPR in↑					AUPR out↑				
Ours	**0.047**	**0.046**	**0.046**	**0.047**	**0.045**	**99.9**	**99.7**	**99.9**	**99.2**	**99.9**	**92.0**	**95.8**	**97.3**	**98.1**	**98.7**
GPND	5.4	5.8	5.8	5.9	6.0	99.7	99.4	99.1	98.6	98.0	86.3	92.2	95.0	96.5	97.5
AE	11.4	11.4	11.6	12.0	12.2	98.9	97.8	95.8	93.2	90.0	78.0	86.0	89.7	92.0	94.0
P-VAE	9.8	9.7	9.7	9.7	9.5	99.3	98.7	97.8	96.7	95.6	81.7	89.2	92.5	94.6	96.3
P-AAE	9.4	9.3	9.5	9.8	9.8	99.2	98.6	97.4	96.0	94.3	79.3	87.7	91.5	93.7	95.4

Table 6. Ablation study that shows F_1 scores for MNIST with various choices of the proposed isometric auto-encoder components.

% of outliers	10	20	30	40	50
Without $\mathcal{L}_{iso\_AE}(f,g)$	0.980	0.960	0.940	0.954	0.910
With $\mathcal{L}_{iso\_AE}(f,g)$	0.984	0.976	0.968	0.960	0.953

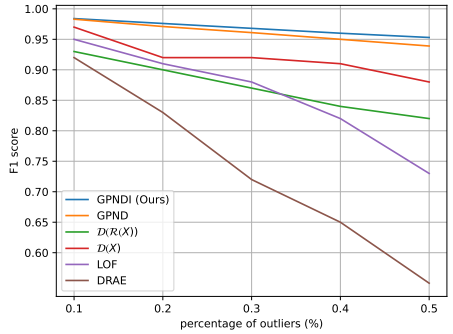

Figure 3. Results on MNIST [23] dataset.

5.4. Ablation

In this section we analyze the contribution of each added part to the encoder that makes it isometric. We investigate how the results change when we include the proposed isometric auto-encoder loss components versus when they are dropped. We repeat the experiment with MNIST as following: without having $\mathcal{L}_{iso\_AE}(f,g)$, and with $\mathcal{L}_{iso\_AE}(f,g)$. Table 6 shows the results of these two settings. The influence of the isometric constraints can be noticed, since the F_1 scores decrease when they are not included.

Moreover, we compare our method, using the MNIST dataset, with [37] and other baselines to better appreciate the improvement provided by the architectural choices. The baselines are: i) vanilla AE with thresholding of the reconstruction error and same pipeline (AE); ii) proposed approach where the AAE is replaced by a VAE (P-VAE); iii) proposed approach where the AAE is without the additional adversarial component induced by the discriminator applied to the decoded image (P-AAE). Table 5 shows that our method exceeds others in all metric measurements.

Other implementation details include the choice of hyperparameters. λ_{iso} is set to 0.01. The hyperparameters λ for $\mathcal{L}_{auto\_error}$ versus $\mathcal{L}_{adv-d_z}$, when optimizing for D_z are equal to 2.5. For MNIST, Fashion-MNIST, and COIL-100 the latent space size was chosen to give the highest F_1 on the validation set which is equal to 16. For CIFAR-10 and CIFAR-100, the latent space size was set to 256. For CIFAR-10 and CIFAR-100, the hyperparameter of λ is 10.0. We use the Adam optimizer with learning rate of 0.0002 for MNIST, 0.00001 for Fashion-MNIST, 0.0003 for COIL-100, CIFAR-10, and CIFAR-100, batch size is 128, and 100 epochs for all datasets.

6. Conclusion

In this work we present GPNDI, an updated framework to GPND [37], where we overcome several weaknesses. By revising the theoretical formulation we motivate the need for using an autoencoder that learns an isometry and a pseudo-inverse map. Those, in turn, preserve the geometric structure of the data, but more importantly, regularize the learning and dramatically simplify the inference model by eliminating the need to compute Jacobians. Extensive experiments demonstrate that the approach based on the proposed new set of losses, learns the manifold of inliers effectively, while reducing the dimensionality of the representation. Novelty detections metrics consistently underscore the increased effectiveness of the revised approach.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1920920.

References

[1] Arslan Basharat, Alexei Gritai, and Mubarak Shah. Learning object motion patterns for anomaly detection and improved object detection. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008. 1

[2] Paul Bodesheim, Alexander Freytag, Erik Rodner, Michael Kemmler, and Joachim Denzler. Kernel null space methods for novelty detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3374–3381. IEEE, 2013. 2

[3] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015. 5

[4] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *ACM sigmod record*, volume 29, pages 93–104. ACM, 2000. 6

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009. 6

[6] Terrance DeVries and Graham W Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018. 2

[7] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016. 2

[8] Eleazar Eskin. Anomaly detection over noisy data using learned probability distributions. In *In Proceedings of the International Conference on Machine Learning*. Citeseer, 2000. 1

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[10] Amos Gropp, Matan Atzmon, and Yaron Lipman. Isometric autoencoders. *arXiv preprint arXiv:2006.09289*, 2020. 4, 5

[11] Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 98–107, 2022. 2

[12] Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K Roy-Chowdhury, and Larry S Davis. Learning temporal regularity in video sequences. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 733–742. IEEE, 2016. 2

[13] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017. 2

[14] Ryota Hinami, Tao Mei, and Shin'ichi Satoh. Joint detection and recounting of abnormal events by learning deep generic knowledge. In *Proceedings of the IEEE international conference on computer vision*, pages 3619–3627, 2017. 2

[15] Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. Neural autoregressive flows. In *International Conference on Machine Learning*, pages 2078–2087. PMLR, 2018. 2

[16] Keizo Kato, Jing Zhou, Tomotake Sasaki, and Akira Nakagawa. Rate-distortion optimization guided autoencoder for isometric embedding in euclidean latent space. In *International Conference on Machine Learning*, pages 5166–5176. PMLR, 2020. 5

[17] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NIPS*, pages 5574—-5584, 2017. 2

[18] JooSeuk Kim and Clayton D Scott. Robust kernel density estimation. *Journal of Machine Learning Research*, 13(Sep):2529–2565, 2012. 1

[19] Ki Hyun Kim, Sangwoo Shim, Yongsub Lim, Jongseob Jeon, Jeongwoo Choi, Byungchan Kim, and Andre S Yoon. Rapp: Novelty detection with reconstruction along projection pathway. In *International Conference on Learning Representations*, 2019. 2

[20] Ivan Kobyzev, Simon J D Prince, and Marcus A Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(11):3964–3979, Nov. 2021. 4

[21] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 6

[22] Steffen L Lauritzen. The em algorithm for graphical association models with missing data. *Computational statistics & data analysis*, 19(2):191–201, 1995. 1

[23] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998. 6, 8

[24] Gilad Lerman, Michael B McCoy, Joel A Tropp, and Teng Zhang. Robust computation of linear models by convex relaxation. *Foundations of Computational Mathematics*, 15(2):363–410, 2015. 7

[25] Shiyu Liang, Yixuan Li, and R Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*, 2018. 2, 6, 7

[26] Guangcan Liu, Zhouchen Lin, and Yong Yu. Robust subspace segmentation by low-rank representation. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 663–670, 2010. 7

[27] Juncheng Liu, Zhouhui Lian, Yi Wang, and Jianguo Xiao. Incremental kernel null space discriminant analysis for novelty detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 792–800, 2017. 2

[28] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33:21464–21475, 2020. 2

[29] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015. 5

[30] HDK Moonesignhe and Pang-Ning Tan. Outlier detection using random walks. In *Tools with Artificial Intelligence, 2006. ICTAI'06. 18th IEEE International Conference on*, pages 532–539. IEEE, 2006. 7

[31] HDK Moonesinghe and Pang-Ning Tan. Outrank: a graph-based outlier detection framework using random walk. *International Journal on Artificial Intelligence Tools*, 17(01):19–36, 2008. 7

[32] Benjamin Nachman and David Shih. Anomaly detection with density estimation. *Phys. Rev. D*, 101(7):075042, Apr. 2020. 4

[33] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*, 2018. 2

[34] John Nash. The imbedding problem for riemannian manifolds. *Ann. Math.*, 63(1):20–63, 1956. 4

[35] Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. Columbia object image library (coil-20). 1996. 6

[36] Jaewoo Park, Yoon Gyo Jung, and Andrew Beng Jin Teoh. Discriminative multi-level reconstruction under compact latent space for one-class novelty detection. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 7095–7102. IEEE, 2021. 2

[37] Stanislav Pidhorskyi, Ranya Almohsen, and Gianfranco Doretto. Generative probabilistic novelty detection with adversarial autoencoders. *Advances in neural information processing systems*, 31, 2018. 1, 2, 3, 4, 5, 6, 7, 8

[38] Mostafa Rahmani and George K Atia. Coherence pursuit: Fast, simple, and robust principal component analysis. *IEEE Transactions on Signal Processing*, 65(23):6260–6275, 2016. 7

[39] Mahdyar Ravanbakhsh, Moin Nabi, Enver Sangineto, Lucio Marcenaro, Carlo Regazzoni, and Nicu Sebe. Abnormal event detection in videos using generative adversarial nets. *arXiv preprint arXiv:1708.09644*, 2017. 2

[40] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015. 2

[41] Oliver Rippel, Patrick Mertens, and Dorit Merhof. Modeling the distribution of normal data in pre-trained deep features for anomaly detection. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6726–6733. IEEE, 2021. 2

[42] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but differnet: Semi-supervised defect detection with normalizing flows. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1907–1916, 2021. 2

[43] Lukas Ruff, Jacob R Kauffmann, Robert A Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G Dietterich, and Klaus-Robert Müller. A unifying review of deep and shallow anomaly detection. *Proc. IEEE*, 109(5):756–795, May 2021. 1

[44] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli. Adversarially learned one-class classifier for novelty detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3379–3388, 2018. 2, 5, 6

[45] Walter J. Scheirer, Anderson Rocha, Archana Sapkota, and Terrance E. Boult. Towards open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 35, July 2013. 1

[46] Seung Yeop Shin and Han-joon Kim. Extended autoencoder for novelty detection with reconstruction along projection pathway. *Applied Sciences*, 10(13):4497, 2020. 2

[47] Mahdi Soltanolkotabi, Emmanuel J Candes, et al. A geometric analysis of subspace clustering with outliers. *The Annals of Statistics*, 40(4):2195–2238, 2012. 7

[48] FYYZS Song and Ari Seff Jianxiong Xiao. Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 6

[49] Yiyou Sun, Chuan Guo, and Yixuan Li. React: Out-of-distribution detection with rectified activations. *Advances in Neural Information Processing Systems*, 34, 2021. 2

[50] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. *Advances*

in neural information processing systems, 33:11839–11852, 2020. 2

[51] Lionel Tarassenko, Paul Hayton, Nicholas Cerneaz, and Michael Brady. Novelty detection for the identification of masses in mammograms. 1995. 2

[52] Manolis C Tsakiris and René Vidal. Dual principal component pursuit. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 10–18, 2015. 7

[53] Huan-gang Wang, Xin Li, and Tao Zhang. Generative adversarial network based novelty detection usingminimized reconstruction error. *Frontiers of Information Technology & Electronic Engineering*, 19(1):116–125, 2018. 2

[54] Yan Xia, Xudong Cao, Fang Wen, Gang Hua, and Jian Sun. Learning discriminative reconstructions for unsupervised outlier removal. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1511–1519, 2015. 2, 6

[55] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 6

[56] Dan Xu, Elisa Ricci, Yan Yan, Jingkuan Song, and Nicu Sebe. Learning deep representations of appearance and motion for anomalous event detection. *arXiv preprint arXiv:1510.01553*, 2015. 2

[57] Huan Xu, Constantine Caramanis, and Sujay Sanghavi. Robust pca via outlier pursuit. In *Advances in Neural Information Processing Systems*, pages 2496–2504, 2010. 7

[58] Pingmei Xu, Krista A Ehinger, Yinda Zhang, Adam Finkelstein, Sanjeev R Kulkarni, and Jianxiong Xiao. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv preprint arXiv:1504.06755*, 2015. 6

[59] Kenji Yamanishi, Jun-Ichi Takeuchi, Graham Williams, and Peter Milne. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Data Mining and Knowledge Discovery*, 8(3):275–300, 2004. 1

[60] Chong You, Daniel P Robinson, and René Vidal. Provable self-representation based outlier detection in a union of subspaces. *arXiv preprint arXiv:1704.03925*, 2017. 7

[61] Jiawei Yu, Ye Zheng, Xiang Wang, Wei Li, Yushuang Wu, Rui Zhao, and Liwei Wu. Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows. *arXiv preprint arXiv:2111.07677*, 2021. 2

[62] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*, 2018. 2